THE *CELTIC INSCRIBED STONES PROJECT* DATABASE: A GUIDE

KRIS LOCKYEAR

REVISED DRAFT: June 1, 2000

Contents

Chapter 1

An introduction to databases

1.1 Introduction

The *Celtic Inscribed Stones Project* (CISP) is jointly run between the Department of History, UCL, and the Institute of Archaeology, UCL, under the direction of Prof. Wendy Davies in collaboration with Prof. James Graham-Campbell. The project currently (as of June 1, 2000) employs three full-time staff (Dr Kris Lockyear, Dr Mark Handley and Dr Paul Kershaw). The database structure described in this manual was constructed with by Dr Kris Lockyear and former research fellow Dr Katherine Forsyth. The first three years of the Project was funded by the HRB/HEFCE via their institutional fellowship scheme.

CISP's aim is to undertake a collaborative, interdisciplinary study of Early Medieval Celtic inscriptions. One of its main objectives is the compilation of a comprehensive and authoritative database of all known inscriptions from Great Britain, Ireland and Brittany. By bringing this material together in one place and making it readily available our goal is to turn what is a largely untapped resource into usable material.

Further details of the Project are available on the Project's web pages (http://www.ucl.ac.uk/archaeology/cisp).

This guide and manual is intended both as a general introduction to the CISP database, and as a detailed guide for data entry. Chapter 1 contains an introduction to databases, database management systems, and data structures (terms which are discussed below). The subsequent chapters discuss the contents of the CISP database, and provide a detailed table by table, field by field guide to the database including allowed terms and definitions of fields and entries, and a short guide to the CISP data entry application. Appendices provide a glossary of terms and list major changes to the database since the first version of this manual.

1.2 Database concepts

This section discusses a number of database concepts and is primarily intended for those who have had little or no experience of computer-based databases.

1.2.1 Databases

A *database* is structured collection of data. Thus, card indices, printed catalogues of archaeological artefacts and telephone directories are all examples of databases. Databases *may* be stored on a computer and examined using a program. These programs are often called 'databases', but more strictly are *database management systems* (DMS). Just as a card index or catalogue has to be constructed carefully in order to be useful, so must a database on a computer. Similarly, just as there are many ways that a printed catalogue can be organised, there are many ways, or models, by which a computerised database may be organised. One of the most common and powerful models is the 'relational' model (discussed below), and programs which use this model are known as *relational database management systems* (RDMS).

Computer-based databases are usually organised into one or more *tables*. A table stores data in a format similar to a published table and consists of a series of rows and columns. To carry the analogy further, just as a published table will have a title at the top of each column, so each column in a database table will have a name, often called a *field name*. The term *field* is often used instead of column. Each row in a table will represent one example of the type of object about which data has been collected. Table 1.1a (p. 6) is a an example of a table from a database of English towns. Each row, in this case a town, is an *entity*, and each column represents an *attribute* of that entity. Thus, in this table 'population' is an attribute of 'town.'

One advantage of computer-based tables is that they can be presented on screen in a variety of orders, formats, or according to certain criteria, *e.g.,* all the towns in Hertfordshire, or all towns with a cathedral.

1.2.2 Specific purpose vs. resource databases

Databases often fall into one of two broad categories. The first comprises specific purpose, limited databases. In academia, these often contain data gathered to perform a relatively limited rôle only in a particular project. The database may be intended to provide the researcher with a particular set of data, but have no particular function or rôle at the conclusion of the project. For example, Lockyear's *Coin Hoards of the Roman Republic* (CHRR) database included only data necessary for the project in hand (Lockyear 1996, chapter 5).

The second category comprises general purpose, resource databases. A good example of a resource database are county archaeological sites and monuments records (SMRs), or national monuments records (*e.g.,* Henrik 1993). These databases are not project specific but are intended to be of use to a wide variety of users. Resource databases usually attempt to be comprehensive within their 'domain of discourse', are maintained and updated, and are made available to interested parties. As these databases attempt to be comprehensive in order to accomodate unpredicted enquiries and research, they include a wide variety of data which in turn requires a complex 'data structure', or way of storing the information.

The CISP database is intended to be a resource database and as a result has a complex data structure (discussed below). This structure, however, provides great power and flexibility both for the retrieval and for the handling of the data, but also for future expansion of the database to include other information and materials.

1.2.3 Relational databases

A common and powerful method for organising data for computerisation is the *relational data model*. Use of this model often results in a database with many tables, and a common question is why such a complex structure should be necessary. Table 1.1b is an example of bad table design with the same towns as in Table 1.1a but with some additional information—the population and the area of the counties—added. We can see from this table that the size and population of Hertfordshire is repeated three times. This duplication is called *data redundency*. Data redundency is a problem for several reasons:

- It is a waste of time to enter the same data repeatedly.
- It increases the possibilities of error. In Table 1.1b the population for Hertfordshire has been mis-typed in the third row.
- Entry errors will create errors in data retrieval, which are likely to be less visible/predictable in complex queries.
- It is a waste of disk space—this can be a major consideration with large databases.
- It can slow down some queries on the database.
- Updates or corrections have to be applied to multiple rows.

A second problem with the table can be seen in the last row. We have information about the population of Essex as a whole but none about any individual town. To accomodate this information we have had to create a row of data with only partial information. As well as these problems, a poor data structure can lead to inflexibility in the use of the database, and possibly problems in retrieving data in the form required. Examples of poor database design are all too common.

To solve these problems, the data should be split into several tables. To follow the town example through, we could have a table of towns as given in Table 1.1a. Each item of information stored in this table is an attribute of a town. The information about counties is then stored in a second, separate table of counties as shown in Table 1.1c. In this table every item of information is an attribute of a county. This process of breaking data down into a series of tables is called *normalisation* and is the first and most important step in designing a relational database.

Normalisation is the process of identifying entities and their attributes, and defining the relationship between the entities. In our example we have two entities—towns and counties, and we have recorded various attributes (Tables 1.1a & 1.1c). There are three types of relationship between entities: *one-to-one*, *one-to-many*, and *many-to-many*. Figure 1.1 shows the different types of relationship in a diagramatic form which are discussed in detail below. This type of diagram is known as an *entity relationship diagram*.

town	county	population	county town?	cathedral?
Welwyn Garden City	Hertfordshire	40,570	no	no
St. Albans	Hertfordshire	123,800	no	yes
Hertford	Hertfordshire	2,023	yes	no
Durham	Durham	29,490	yes	yes

(a) A table of English towns

town	county	population	county town?	cathedral?	county population	county size
Welwyn Garden City	Hertfordshire	40,570	no	no	937,300	631
St. Albans	Hertfordshire	123,800	no	yes	937,300	631
Hertford	Hertfordshire	2,023	yes	no	397,300	631
Durham	Durham	29,490	yes	yes	132,681	295
	Essex				1,426,200	1,528

(b) A badly designed table

county	population	size (square miles)
Hertfordshire	937,300	631
Durham	132,681	295
Essex	1,426,200	1,528

(c) A table of counties

Table 1.1: Example tables from a geographical database

One-to-one relationships

This is where there is, for any one entity, only one example of another related entity. For example, if we had only collected data about *county* towns, there would be a one-to-one relationship between each entry (county) in the table of counties and a town in the table of county towns. This type of relationship is shown in Figure 1.1a. It would be possible, although not really desirable, to store all the information in one table in this case.

A special case of a one-to-one relationship is where particular pieces of information only exist, or are only applicable, to some of the entries in a table. In our geographical example we may wish to record the length of coast line or other attributes which only relate to counties which border the sea. In these cases one can create a separate table for this information. This helps to save disk space on the computer, minimise data entry time, and break down potentially large tables. This type of relationship is shown in Figure 1.1b.

One-to-many relationships

This is where there is, for any one entity, many examples of another entity. This is the relationship between the counties as shown in Table 1.1c and the towns in Table 1.1a—a town can only have one county *but* a county will have many towns. In these cases, the information about each entity must be stored in separate tables. This type of relationship is shown in Figure 1.1c.

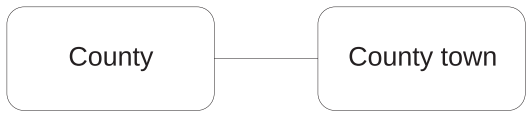
Many-to-many relationships

This is where an entity can have many examples of another entity *but* this second entity can also have many examples of the first. In our geographical example, we may want to store information about rivers. Any one county has many rivers, but similarly, a river is likely to flow through many counties. This type of relationship is illustrated in Figure 1.1d.

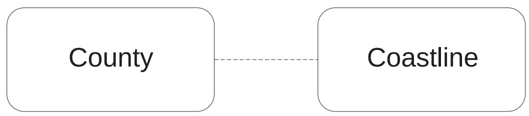
This type of relationship necessitates the use of the third table. This effectively creates two one-to-many relationships. These intermediate tables can be called *linking tables*. These tables often only contain two columns which act as a link between the two main tables. In our geographical example, the linking table would contain the names of counties, and the names of rivers only. This solution to modelling many-to-many relationships is illustrated in Figure 1.1e.

Primary and foreign keys

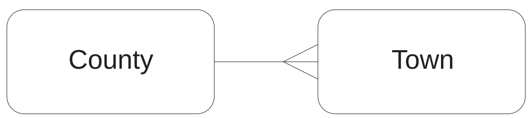
Every row in a table in a relational database must be unique, *i.e.,* there must not be two identical rows. One or more columns are therefore designated the *primary key* (sometimes called the unique identifier) for the items contained within it. Thus, in Table 1.1a the column ‘town’ could act as the primary key, and in Table 1.1c column ‘county’ can act as that table’s primary key. This concept has been used in paper-based (*i.e.,* published) databases, *e.g.,* each inscribed stone catalogued in R. A. S.



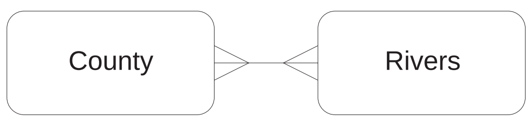
(a) A one-to-one relationship



(b) A one-to-one relationship for some entities only



(c) A one-to-many relationship



(d) A many-to-many relationship

(e) Splitting a many-to-many relationship into two one-to-many relationships

Figure 1.1: Entity relationship diagrams.

Macalister's *Corpus Inscriptionum Insularum Celticarum* (1945, 1949) has an unique identifying number, as does each hoard in Crawford's *Roman Republican Coin Hoards* (1969).

In our geographical example, however, there can be more than one town with the same name, Newcastle or Newport for example. In this case we could designate the 'town' *and* the 'county' columns together as the primary key.

Foreign keys are columns in a table which provide a link to another table. In our geographical example, the county column in our table of towns provides a link to the table of counties, and is thus a key field in that relationship. It is very important therefore to ensure that entries in the both tables are identical, *i.e.,* that both tables use the full county name (Hertfordshire) or an abbreviation (Herts) but not a mixture of the two.

There is one final complexity which must be addressed. What could we do in the case where there are two towns with the same name in the same county? Although in our example it is unlikely, in databases of other information this could happen. We could use a combination of name, county and population as the primary key for the table of towns. If we had a table of shops, we would have to include the town name, county *and* the population to provide a link between the two tables. This, however, will re-introduce the problem of data redundancy. A better course of action is to assign a unique code to each town, and to use this code as the link to the table of shops. The use of codes has other advantages: it can be quite short and thus save time during data entry and disk space. These codes can be assigned by the user, *e.g.,* WGC for Welwyn Garden City, or could be a sequential number created automatically by the program.

Data types and definition

The data stored in tables can be classified into types. In Table 1.1a the first column can contain any letter, number, or other character (such as {, or &). This is an alphanumeric data type, also known as a string or character field. The third column for population contains a number and is a numeric data type. The last two columns are 'logical' and can only contain yes or no. There are other data types such as date or even images and sounds.

The type of data is important as different types of data behave in different ways. A good example is the sorting order of a series of numbers. If we store 1, 22, 3, 10, 2 and 15 in a numeric column, and ask the program to sort the rows of the table on this column, we will get 1, 2, 3, 10, 15, 22 as we might expect. If that column was defined as an alphanumeric data type, the result would be 1, 10, 15, 2, 22, 3, *i.e.,* a rather different result! Different DMSs have different ways of handling different types of data (see below).

Each column of data also has to be defined. This can be quite simple, *e.g.,* 'the county column will contain the full county name'. We also have to decide what the entries mean, *e.g.,* in the table of counties we have a column for area—we have to decide if this is the area in square miles or square kilometers.

We may wish to restrict the possible entries in a column. We can do this to prevent errors, *e.g.,* we may decide that the maximum allowed population in a town is 10,000,000 as no town in Britain has a population larger than that. We may also wish to restrict entries to a limited list of terms. If, for example, we had ‘type’ as an attribute of town, we could have market town, small town, county town, village, small village, hamlet and so on. If any term was allowable, this attribute would not be very useful for retrieving groups of settlements in any meaningful way. We might, therefore, create a list of allowed terms which are precisely defined and which would therefore allow meaningful data retrieval.

Look-up tables

In the previous section restricted data fields were discussed. How, in practice, are the entries in fields to be restricted? The first method is for the allowed terms to be listed in a manual such as this one, and for every user to be disciplined enough only to use those terms, and to check that they have used the correct ones. There are advantages, however, in storing these terms on the computer along with the main tables of data. There are thus two further methods. The first is to include the definitions in a database application (see below), or in the way the table is defined within the DMS. This has the disadvantage that the information is dependent on the software being used, and if the data is transferred (‘ported’) to another program this information will be lost. It is also difficult to add new terms to the list. The second alternative method is to use *look-up tables*, of which there are two types, simple and hierarchical.

Simple look-up tables typically consist of one or two columns. In a one column example, the list of allowed terms is stored in the table; in a two column example the first column stores the allowed term, often in the form of a code, and the second column stores the definition of that term or code. A good example of simple look-up tables are the POSIT1, POSIT2 and POSIT3 tables discussed on page 40.

Hierarchical look-up tables are very similar in that one column contains a series of unique terms or codes. The remaining columns then contain definitions of that code, but in different levels of detail. Using our geographical example, we might wish to classify the rivers. The look-up table would contain a column of codes. Another column could then contain some broad classification such as ‘major river’, ‘minor river’ and ‘stream.’ A third column could then further subdivide the classification, *e.g.,* major rivers might be divided into ‘tidal’ and ‘non-tidal’, and a fourth column could divide ‘tidal’ into ‘estuarine’ and ‘non-estuarine’. The SITETYPE table discussed in section 3.2.6 is a good example of a hierarchical look-up table.

Hierarchical look-up tables have a dual function—to restrict the entries in a second table (sometimes called a parent table), and to provide a mechanism by which complex queries can be simplified. Both types of look-up table can be used to create printed output from the database which is more readily understood, by replacing a series of possibly obscure codes with more descriptive pieces of text.

Database applications

Relational database management systems (RDMS) will typically provide a series of tools for creating tables, conducting searches, producing printed reports, *etc.* With a complicated database, however, it is usual for a *database application* to be written. A database application is a usually a program within a program, *i.e.,* it is a program that runs inside the RDMS. Most, if not all RDMSs, provide an ‘application development language.’ This will allow a computer programmer to create an application to perform specific tasks for a particular database, most commonly to provide a simpler and more efficient method of inputting data to the database, and for checking for errors. Often this will use a series of forms with menus and buttons.

1.3 Conclusions

This chapter has provided an overview of the concept of databases, and has presented detail relating to the concept of relational databases, their structure and requirements. For those wishing to go further the database Bible remains Date’s *An Introduction to Database Systems* (Date 1995); Carter (1992) provides a less comprehensive but perhaps more comprehensible account for non-specialists.

The following chapters examine the content and structure of the CISP database in general, and then provide a data definition guide to all tables and fields.

Chapter 2

The CISP database

This chapter firstly defines the 'domain of discourse', *i.e.,* the information to be recorded in the database, and the discusses the structure of the database in the light of the concepts discussed in the previous chapter. Throughout this guide the following conventions are used: table (relation) names are given in SMALL CAPITALS, field (or column) names are given in a `fixed space font` and allowed entries for a restricted field in **bold type**.

2.1 Criteria for inclusion

The database is to include all 'early medieval inscribed stone monuments from Celtic-speaking regions' defined as follows:

- 'early medieval'
  - lower limit — earliest non-Roman inscriptions in the region
  - upper limit — *c.* 1100. This may need to be further defined on a region-by-region basis. Since not all stones are closely datable it may be necessary to decide to include all examples of a particular sub-type (details to be decided later, but inscriptions in Lombardic or 'Black' lettering definitely excluded; Romanesque probably excluded).
- 'inscribed'
  - carved with linguistic text (including 'alpha-and-omega' but excluding 'chi-rho monogram')
  - exclude stones carved with crosses and/or other ornament but no text
  - exclude Pictish symbol stones
- 'stone monuments'
  - exclude all inscriptions on materials other than stone
  - exclude inscriptions on stone objects other than monuments or buildings (*e.g.,* stone lamps, stone whorls)
- 'Celtic-speaking regions'
  - all of Ireland, Scotland, Wales, Isle of Man, Dumnonia, Brittany but exclude:
    - stones inscribed solely in runes (Anglian or Scandinavian)
    - Roman period Celtic (Gaulish) inscriptions from the Continent

    * unambiguously Anglo-Saxon inscriptions from southern Scotland and the Welsh and Dumnonian marches
- include inscriptions from elsewhere in England and France and the Channel Islands which are written in a Celtic language or are otherwise of distinctively 'Celtic' type

Rule: If in doubt, include — it is better to include things not strictly within the remit than to omit things that should be covered.

2.2 Doubtful inscriptions and lost stones

N.B. 'Lost' refers to stones not inscriptions. If the stone is extant but the inscription is no longer visible (whether through extreme wear or because the inscribed portion has been broken and lost) it is 'doubtful' only if the extant descriptions and illustrations are unreliable.

'Lost' stones are those which have been described and/or illustrated by a credible authority but whose current whereabouts are unknown (*i.e.,* stones which have been destroyed, stolen, or mislaid). They are described as normal and the current location is given as 'lost'. The dating parameters for their loss are noted (see the LOST table below).

If there is doubt that the inscription on a lost stone was genuine, for instance if the description is ambiguous or fantastical, then the stone and its inscription are listed as normal and 'doubtful' is recorded in the inscription table. The same goes for extant stones if there is doubt that visible scores (a) are artificial, (b) represent lettering, or (c) date to the early medieval period.

2.3 The structure of the CISP database

The CISP database is comprised of a primary group of five tables, each in a one-to-many relationship with another table in a hierarchy. These tables form the core of the database. As it is an explicit aim of the project to examine these monuments in their wider context, the first primary table is the SITE table. Each site is given a primary key in the form of a five letter code. Any one site, however, can have more than one stone—there is a one-to-many relationship between sites and stones. The stones are, therefore, recorded in a separate table (STONE), and linked via the site's primary key. Similarly, any one stone can have more than one inscription and thus we have another one-to-many relationship and a third table (INSCRIP). Similarly, any one inscription can have more than one reading, and an authority can provide us with more than one interpretation of a reading. This gives us the fourth and fifth tables (READING & TRANSLAT).

For each of these primary tables there are some related subsidary tables. For example, a site may be known by more than one name. CISP will record the most commonly used name in the SITE table. Alternative names, if any, are stored in a separate table, ALT_NAME. This allows a site to be accessed using any of the possible site names.

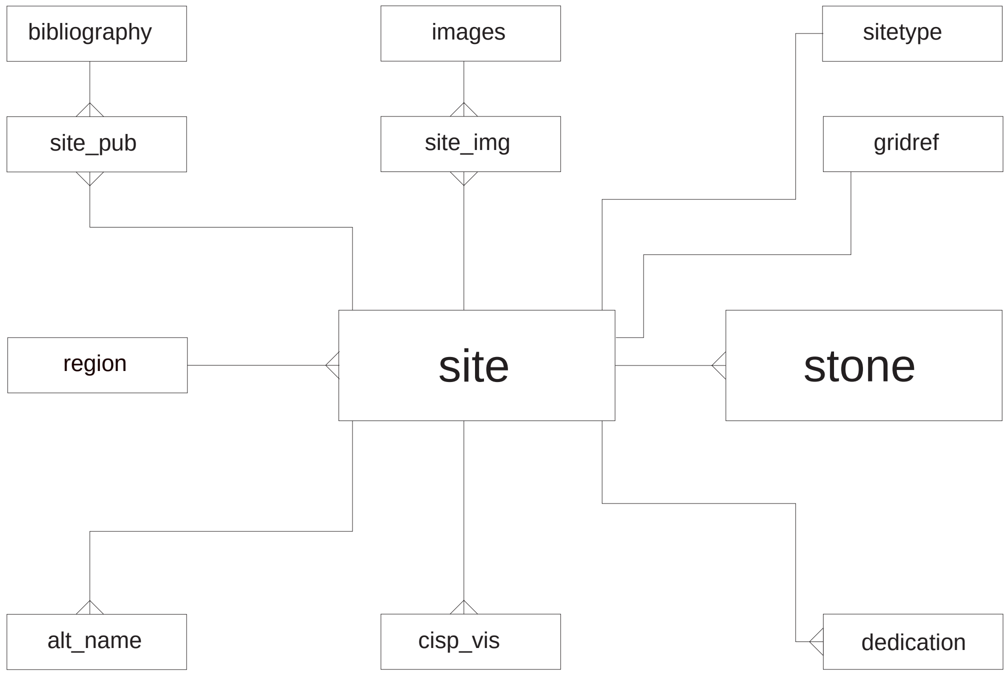


Figure 2.1: Tables related to the primary SITE table

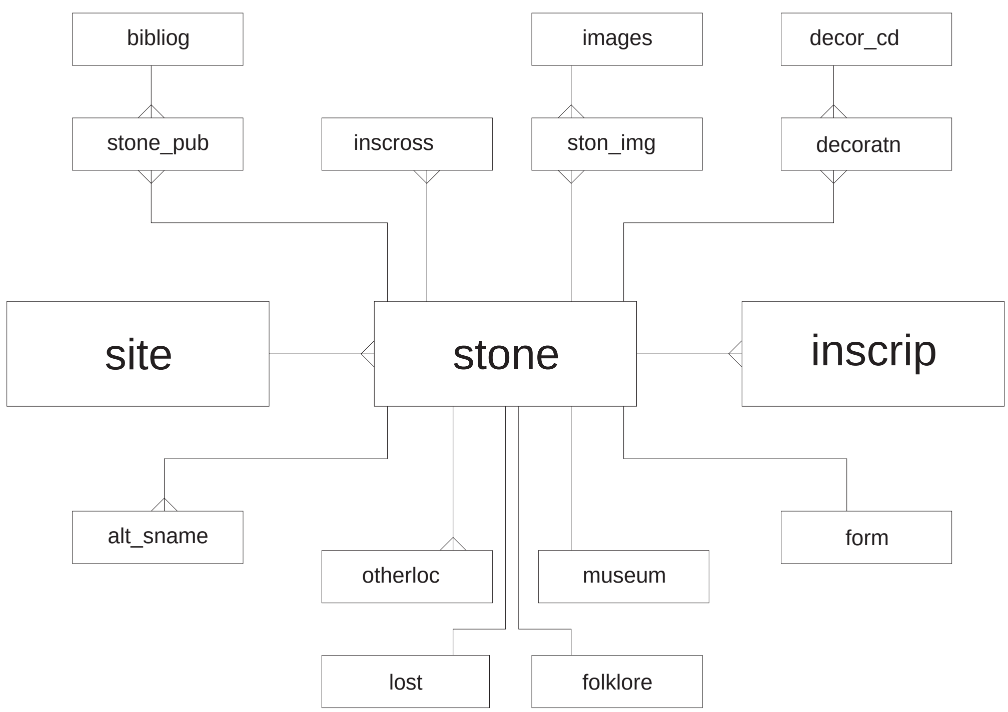


Figure 2.2: Tables related to the primary STONE table

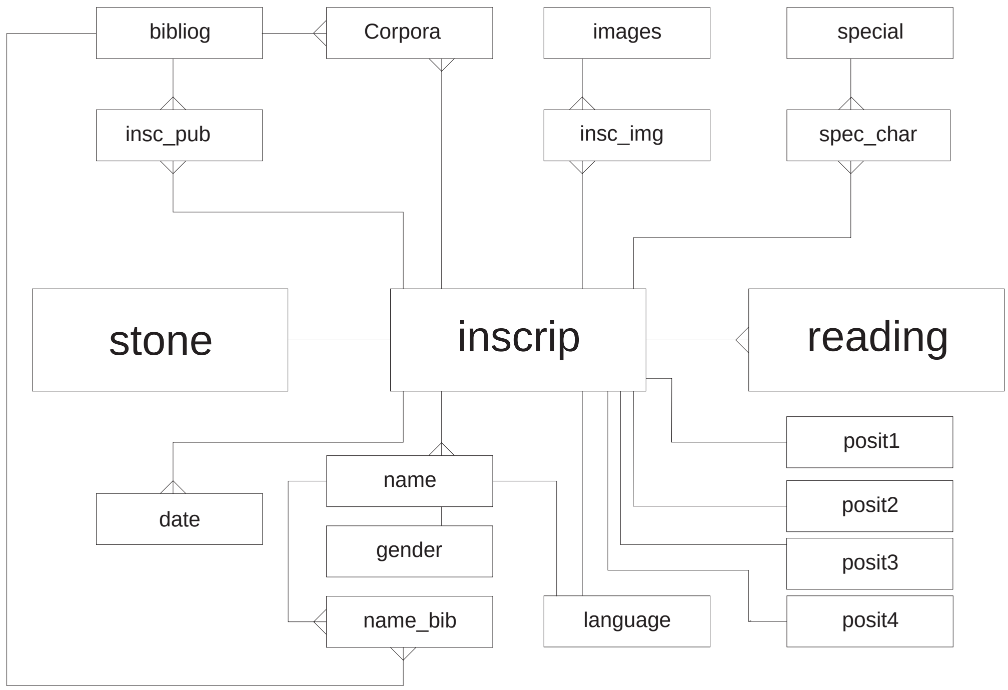


Figure 2.3: Tables related to the primary INSCRIP table

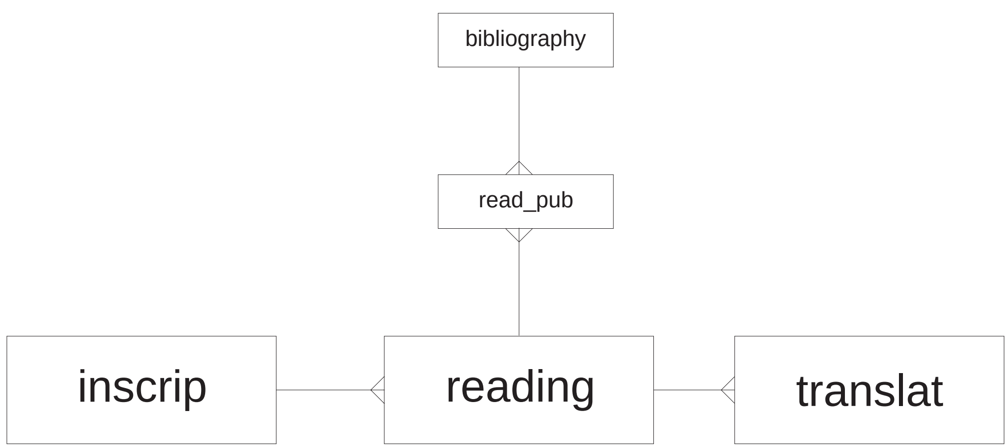


Figure 2.4: Tables related to the primary READING and TRANSLAT tables

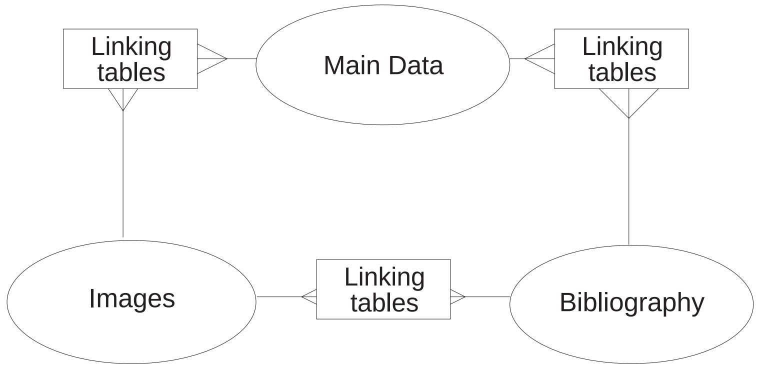


Figure 2.5: General structure of the CISP database

Figures 2.1–2.4 show the five main tables, and their relationship to the other primary tables, and their subsidiary tables.

In addition to these five main tables and the subsidiary tables, there are a large number of simple and hierarchical look-up tables. These perform the functions discussed above (see page 10); all are listed in the next chapter and the hierarchical tables are discussed in detail.

We have, therefore, five primary tables which have a set of related secondary tables and look-up tables. These tables form the core of the CISP database. There are a number of other tables — it is helpful to divide the database into three groups or subsystems (see Figure 2.5). These are the core, bibliography and image subsystems.

The bibliography subsystem has one primary table, BIBLIOG which will contain all the references to published sources and systematic archives such as sites and monuments records. This table is connected to four of the primary tables via a series of linking tables (see page 7). It is also connected to the NAME table and the IMAGE table. These linking tables contain, as well as the necessary linking information, further data such as specific page numbers, and an indication of the value of the reference, *e.g.,* 'major discussion' or 'passing mention'. Finally, the CORPORA table forms an alternative link between the INSCRIP and BIBLIOG tables and allows the retrieval of inscriptions by their standard reference numbers such as those in Macalister (1945, 1949). Figure 2.6 illustrates the relationships.

The IMAGE table stores images of sites, stones and inscriptions. This table has a many-to-many relationship with three of the primary tables and the bibliography table, and is therefore linked to them via intermediate linking tables. Lastly, the CISPARCH table records information about the CISP archive including photographs, rubbings and correspondence. This table is linked to the SITE, STONE, and INSCRIP tables via further linking tables. Figure 2.7 illustrates these last relationships.

Although this data structure is complex, it allows for flexible data retrieval, and for future expansion of the project. For example, no changes to the structure of the database would be required in order to add runic and Anglo-Saxon inscriptions to the system, although some of the terms in the look-up tables might well need expansion. Portable artefacts could be added by creating an artefact table in parallel to the STONE table,

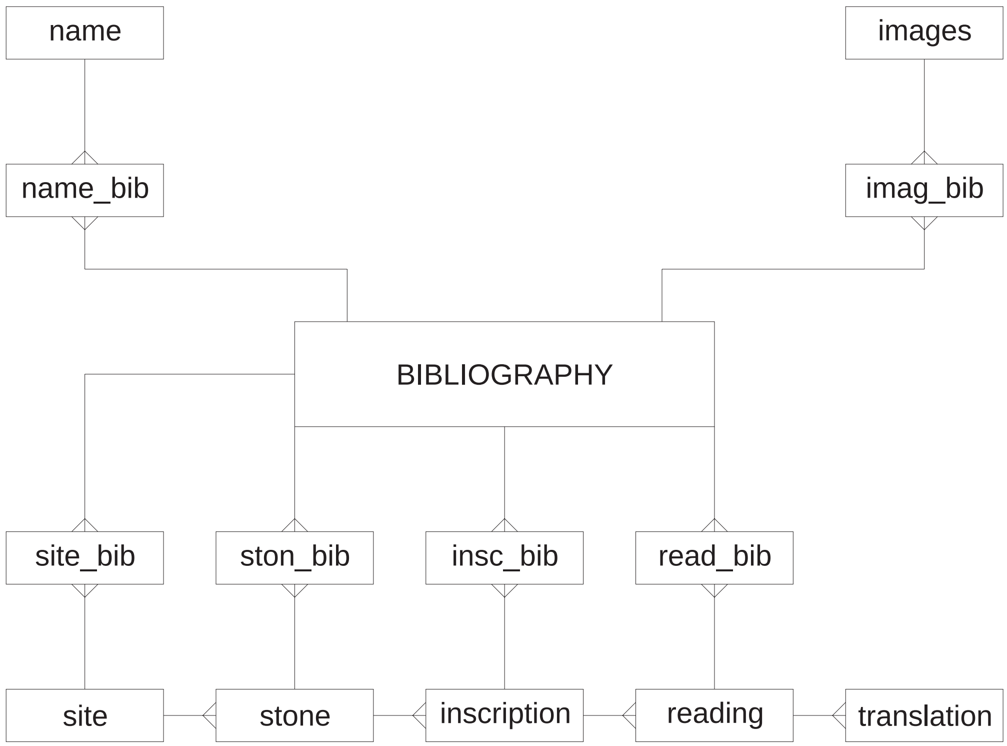


Figure 2.6: The bibliographic subsystem

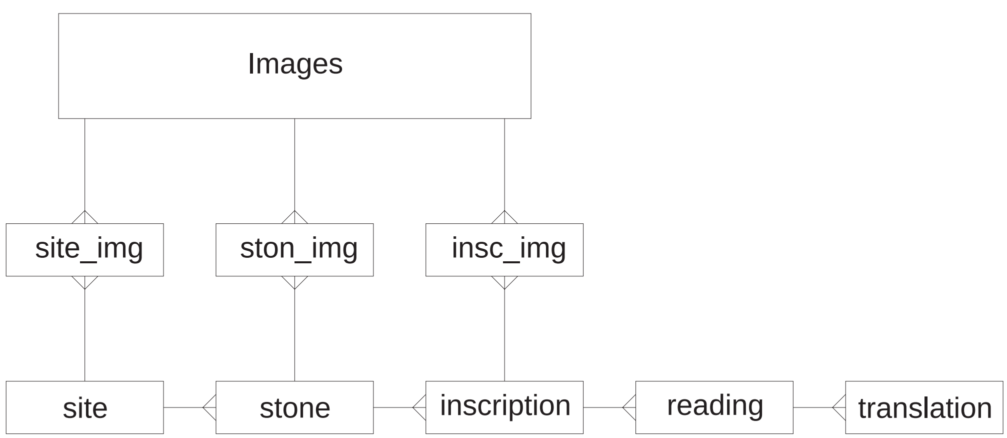


Figure 2.7: The image subsystem

or more detailed recording of decoration or form could be added by creating further tables also linked to the STONE table.

2.4 Data definition strategy

Tables in the CISP database consist of three basic types of fields. The first group are relatively short but unrestricted fields such as place names, measurements, or readings of inscriptions. The second group are fields with a very restricted range of entries controlled by look-up tables. The final group of fields are memo or very long string fields. Database purists dislike this type of field as it is unstructured and can lead to poor efficiency in data retrieval. In the case of the CISP database, it was felt that there were many cases where free form text was necessary, and an even more complex data structure would be counter-productive. In these situations, some information is stored in a structured format in restricted fields, and some in a memo format. For example, the STONE table has two logical (yes/no) fields, `cross` and `other_carve` which record whether a stone has an inscribed cross and/or other carved decoration. If the answer is yes, further structured details are stored in the INSCROSS and/or DECORATN tables, and a fuller, free text description is entered into the DECOR_NOTES memo field.

2.5 Database implementation

2.5.1 Hardware and software

The above data schema is a theoretical construct derived from the information CISP wishes to record. This schema is independent of software or hardware platforms, *i.e.,* it can be implemented using different RDMSs on different types of computer. As a result, the database can be moved ('ported') from one computer package to another with relatively few problems.

The primary obstacle to data transfer between systems is that different packages have different methods for storing data. For example, dBASE and PARADOX store long text (memo) fields in a separate computer file. These memo fields are of unlimited length whereas other strings are limited to 256 characters. Microsoft ACCESS can store strings up to 64kbytes as part of the main table.

The database was originally implemented using *Visual* dBASE 5.5 on a Pentium 133mhz PC.[1] One copy of the *Visual* dBASE client–server bundle of computer programs was purchased which includes a program to help ease data transfer between different database systems.

2.5.2 The data entry application

The complex nature of the CISP database necessitated the construction of a data input application which was written by the author using *Visual* dBASE's 'object orientated'

1. Technical note: it was decided to use Paradox format tables within *Visual* dBASE in order to take advantage of the facilities to ensure primary keys and referential integrity.

programming language.[2] Currently, only a single site/stone *etc.* addition/browsing capability has been implemented. This application speeds up data entry, and helps minimise errors by the automatic creation and linking of primary keys wherever possible. The application allows the user to easily navigate around related sets of tables.

2.5.3 Database dissemination

It has been decided to mount the first release of the database on UCL's Oracle database server which will be accessible via a relatively simple database application accessible through the Web.

No decisions have been made as to the method by which the full and final version of the CISP database is to be disseminated. This is to keep the options open given the rapid development of computer technology, particularly the Internet. There are a number of possible options currently available, none of which need be exclusive.

- Submission of the database to the archaeology section of the Humanities Data Archive currently based in York. This is a requirement of the Project's funding.
- Dissemination of the data on disk, most probably on a CD-ROM. It would be possible to provide a compiled version of the database application (*i.e.,* a version of the application which did not require the user to have *Visual* dBASE) as well as the data.
- Providing an interface to the database via the World Wide Web.

2. Technical note: one should be clear that *Visual* dBASE is a relational database management system with an OOP application language, not an OO DMS.

Chapter 3

A guide to the CISP database tables and fields

This chapter describes each table and field in turn and in detail along with allowed entries, meanings and other relevant information. Simple look-up tables will not be discussed in detail. After a few general points, each of the five primary tables are discussed first along with related subsidiary and hierarchical look-up tables, followed by the bibliographic, archive and image subsystems.

3.1 General guidelines

Fields marked with a dagger (†) are memo fields, fields marked with § are the table's primary key fields.

As a general rule, no entries are to be left blank. If the field is not applicable **n/a** should be entered; if information is not available, **inc** (*i.e.,* 'incomplete') should be entered; if the information will never be available (for example the details of an inscribed cross are too warn to be described) then **ind** (*i.e.,* indeterminate) should be entered.

Memo fields should conform to the following guidelines.

- Memo fields contain free-form text but the emphasis shall be on quoting verbatim from the literature with appropriate bibliographic references. The references should be of the following form:
  - `Macalister/1945: '...'` *or*
  - `Macalister/1945, 345: '...'` *or*
  - `Macalister/1945, 360--361: '...'`

  where the form of the reference is the same as that of the primary key in the BIBLIOG table. Note the use of the correct opening quote (`'`) and the double dashes in the page ranges. In many cases a specific page reference will be necessary.
- The information in the memo fields should be given in a concise note form.
- *Italics* in memo fields should be indicated by underscores, *i.e.,* `_some italic text_`.
- Paragraphs should be indicated by a blank line.

3.2 The SITE and related tables

Tables considered in this section are SITE, REGION, ALT_NAME, GRIDREF, SITETYPE, and SAINT. Tables SITE_PUB and SITE_IMG are discussed in sections 3.7–3.8.

3.2.1 The SITE table

The ‘site’ is the earliest known location of a stone, not necessarily its current location.

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
2	`Name`	alphanumeric	30
3	`Cel_name`	alphanumeric	30
4	`Place`	alphanumeric	30
5	`Cel_place`	alphanumeric	30
6	`Parish`	alphanumeric	30
7	`Cel_parish`	alphanumeric	30
8	`County`	alphanumeric	30
9	`Cel_county`	alphanumeric	30
10	`Country`	alphanumeric	30
12	`Site_type`	alphanumeric	5
†13	`Site_descrip`	memo	–
†14	`Site_history`	memo	–

1. Unique five letter alphanumeric code in capitals. Code is derived from the site’s name and is assigned by CISP, and acts as the primary key for the table.
2. Name of site. Use the name current in recent scholarly literature (list alternative names in ALT_NAME table). In most cases this will be the name of an adjacent farm, church or village, or the name of the parish, townland or island. Despite the inconsistencies, names used by previous writers are retained unless there is good reason to change (*e.g.,* to differentiate two sites with the same name).
3. The above name in the local modern Celtic language.
4. Name of nearest town or village (in Ireland list the townland). In many cases this will be the same name as fields 2–3.
5. The above name in the local modern Celtic language.
6. Name of ecclesiastical (not civil) parish (in Ireland list barony; in Brittany list commune).
7. The above name in the local modern Celtic language.
8. County (pre-1974 counties for Britain, current counties for Ireland, départements for France)
9. The above name in the local modern Celtic language, if appropriate.
10. Country: **Scotland**, **Ireland** (not distinguishing between the Republic and Northern Ireland, but see REGION below), **Wales**, **England**, **Isle-of-Man**, **Channel-Islands**, **France**.
11. The `site_type` is stored as a five letter lower-case code the meanings of which are stored in the hierarchical look-up table SITETYPE discussed on page 24.

12. †`Site_descrip`: fuller description of extant physical features of site, other standing remains, crop marks, finds. Particular emphasis on early medieval period, but also on earlier phases, later only if deemed relevant (*e.g.,* Romanesque church).
13. †`Site_history`: any relevant historical information, including place-name, documentary refs. to site, including details of any excavation *etc.* Details of dedication or other associations with saint.

In practice, it proved very difficult to split `site_descrip` and `site_history`, and for many sites there was very little information. In the longer term these two fields are likely to be merged.

3.2.2 The SAINT table

The name of any saint associated with site, *e.g.,* place-name, church dedication, historical reference *etc.*. This is stored in a separate table since a site may have more than one, either simultaneously or temporally.

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
§2	`Saint`	alphanumeric	30

1. §Site code; links to SITE table.
2. §Name of saint (without ‘St’).

3.2.3 The REGION table

This table allows otherwise cumbersome searches by historic region. Further regions can be added easily as the need arises. Below are some suggestions.

Field	Field Name	Type	Length
§1	`County`	alphanumeric	30
§2	`Region`	alphanumeric	30

1. §County
2. §Region

Ireland Munster, Leinster, Ulster, Connacht; also: Northern Ireland
Scotland Pictland, Dál Ríada, Between-the-Walls
England Dumnonia
Wales 1974 counties.
France historic Brittany, modern administrative Brittany, diocese.

Munster Kerry, Cork, Waterford, Limerick, Tipperary, Clare
Connacht Galway, Mayo, Sligo, Roscommon, Leitrim
Leinster Wexford, Kilkenny, Carlow, Laois, Wicklow, Offaly, Kildare, Dublin, Meath, Louth, Cavan, Longford, Westmeath

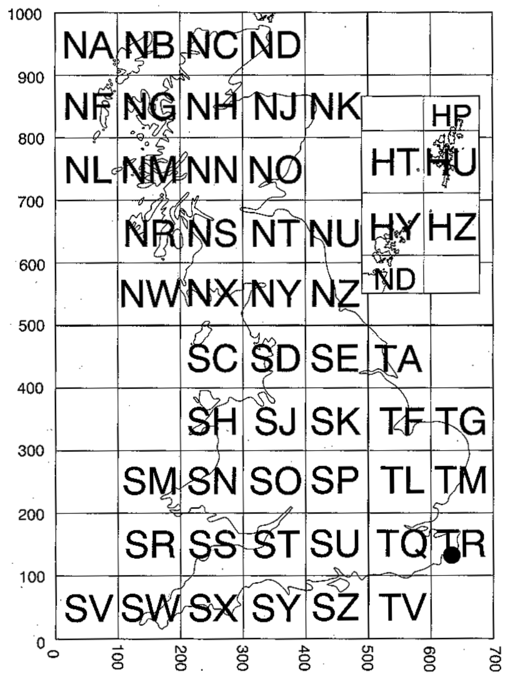


Figure 3.1: Conversion of OS grid letters to numbers. For example, SS 123 456 becomes 212300 145600.

Ulster Antrim, Down, Armagh, Tyrone, Fermanagh, Donegal, Londonderry, Monaghan

Dál Ríada Argyll and Southern Hebrides

Pictland FIF, CLK, PER, ANG, STL, KCD, ABD, BNF, MOR, SUT, CAI, INV, ORK, SHE, ROS, KNR, NAI [replace with full county names].

Between-the-Walls LNK, RNF, AYR, DMF, WIG, BWK, PEB, DNB, ELO, MLO, ROX, WLO (*i.e.,* the British kingdoms between the Roman walls, Strathclyde, Gododdin, Cumbria *etc.*) [replace with full county names].

3.2.4 The GRIDREF table

National Grid Reference given to monument or centre of site. N.B. To ensure compatibility of grid references all must be stored as 12 figure numbers and, therefore their true accuracy must also be recorded (12 figures implies an accuracy to the nearest metre). This is recorded as a measurement in metres, *i.e.,* a GB letters + 8 figure reference (*e.g.,* SS 1456 3256) is to 10 metres, 6 figures to 100 metres, and 4 figures to 1km. The GB letters need to be converted to numbers representing their 100km grid square. For the north of Scotland this will result in a 7 figure northing.

1. §The site code.
2. System (**GB**, **Ireland**, **France**)
3. Eastings: 6 figure reference to include numeric version of letter code.
4. Northings: as above (7 figures for the north of Scotland).
5. Accuracy in metres.

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
2	`System`	alphanumeric	7
3	`East`	numeric	–
4	`North`	numeric	–
5	`Accuracy`	numeric	–

3.2.5 The ALT_NAME table

Alternative name(s) for a site, *e.g.,* Inis Cealtra is also known in the literature as 'Holy Island' and 'Inishcaltra', Toureen Peakaun is also known as 'Kilpeakaun'. See also the ALTSNAME table.

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
§2	`Alt_name`	alphanumeric	40

1. §The site code.
2. §The alternative name.

3.2.6 The SITE_TYP table

This is a look-up table for the `site_type` field of the SITE table.

Field	Field Name	Type	Length
§1	`Sitetype`	alphanumeric	5
2	`Description`	alphanumeric	25

1. §A five letter code in lower case.
2. A description of the site type.

The following lists the contents of this table. It is intended that this list is comprehensive but it may be added to if necessary. The `sitetype` field contains the allowed codes for the `site_type` field of the SITE table.

`Sitetype`	`Description`
cemet	cemetery**[1]**
eccle	ecclesiastical**[2]**
inc	incomplete information
ind	indeterminate
lands	landscape setting
modsc	modern secondary**[3]**
n/a	not applicable
other	other
settl	settlement
soutn	souterrain

[1] *i.e.,* without associated church

[2] *i.e.,* church, abbey, early Christian site, including graveyard

[3] First found re-used in a post-medieval context or structure (other than a church), *e.g.,* as a gatepost, a house lintel, or in a wall.

3.3 The STONE and related tables

Tables considered in this section are the STONE, ALTSNAME, MUSEUM, LOST, OTHERLOC, FOLKLORE, INSCROSS, DECORATN, DECOR_CD, and FORM tables. Tables STON_PUB and STON_IMG are discussed in sections 3.7–3.8. Look-up tables CIRCUM, DAMAGE, SETTING, LOCATION, and STATUS are not discussed in detail.

3.3.1 The STONE table

Field	Field Name	Type	Length
§1	Site	alphanumeric	5
§2	Stone	numeric	–
3	Disc_circ	alphanumeric	15
4	Disc_when	numeric	–
5	Disc_who	alphanumeric	20
†6	Subsq_hist	memo	–
7	Current_setting	alphanumeric	10
8	Current_location	alphanumeric	15
†9	Loc_notes	memo	–
10	Mon_form	alphanumeric	5
†11	Form_notes	memo	–
12	Completeness	alphanumeric	10
13	Preservation	alphanumeric	5
†14	Prescondnotes	memo	–
15	M_sources	logical	–
16	M_height	numeric	–
17	M_ht_status	alphanumeric	10
18	M_width	numeric	–
19	M_wdth_status	alphanumeric	10
20	M_thickness	numeric	–
21	M_thck_status	alphanumeric	10
22	I_source	alphanumeric	25
23	I_ht_feet	numeric	–
24	I_ht_inches	numeric	–
25	I_ht_status	alphanumeric	10
26	I_wd_feet	numeric	–
27	I_wd_inches	numeric	–
28	I_wdth_status	alphanumeric	10
29	I_th_feet	numeric	–
30	I_th_inches	numeric	–
31	I_thck_status	alphanumeric	10
32	No_carved	alphanumeric	2
†33	Decor_notes	memo	–
34	Visited	logical	–

1. §The site code.
2. §Number assigned by the CISP data entry application (if there is more than one

stone at a site, follow traditional numbering where possible).

3. Circumstances of discovery. These codes are stored in the CIRCUM table: **first mentioned** (earliest reference in print or archive, circumstances of discovery unknown), **arch excav** (archaeological excavation), **non-arch dig** (non-archaeological excavation, *e.g.,* laying pipes, digging graves), **recognised** on standing monument (*i.e.,* stone previously known but inscription not recognised as such), found built **in/on structure** or wall (*e.g.,* when it was pulled down, *i.e.,* stone not previously known).
4. Date of discovery or, if unknown, first publication.
5. Name of person who discovered stone or, if unknown (or stone was 'always known'), who published the first account. Names should be entered surname first, *i.e.,* Macalister, R. A. S. If unnamed, phrases like 'workmen' are admissable.
6. †Notes on subsequent history of stone, *e.g.,* later reuse, removal to other site(s), falling down, re-erection, if once part of named antiquarian collection.
7. The current setting of the stone: **unattch** (unattached), **in ground**, **on ground**, **in struct** (in structure), **in display** (default for museums). Values stored in look-up table SETTING.
8. Current location: **earliest** (still in the earliest recorded location); **lost**; **museum**; **on site** (generally on the same site as found, but not in the exact original location, *e.g.,* found originally when digging a grave, now set up in the churchyard); **other** (any other location, for current location see OTHERLOC table). These codes are stored in the LOCATION look-up table.
9. †Precise description of location and related notes (to enable some one looking for it to find it or differentiate it from other stones, also as a bench-mark in case of future movement or loss).
10. Physical form of monument: standard terms only permitted — see description of the FORM table below.
11. †Any further details of the physical form of the stone (excluding ornament).
12. Degree of completeness: **complete** (roughly 90–100% of original monument survives), **incomplete** (60–90%), **frgmntry** (less than 60%). Values stored in the COMPLETE look-up table.
13. State of preservation: **good**, **some-**damage, **poor**. Values stored in the DAMAGE look-up table.
14. †Notes on the present condition if further description is necessary, *e.g.,* 'damaged by recutting for use as building stone', 'water-worn', 'broken at top', 'deteriorated considerably in recent past — cast in NMI made in 1893 shows more detail'.

Measurements

The maximum height, width and thickness of the stone are recorded. These are defined as follows:

Height Vertical axis of upright stones, long axis of recumbents (as per original orientation where this can be determined, not as it now lies if fallen). This measurement will usually be the longest of the three

Width Maximum dimensions of the cross-section, *i.e.,* at 90 degrees to height.

Thickness At 90 degrees to both of the above. This will usually be the shortest measurement of the three.

If additional measurements are required, *e.g.,* because the shape is complex, these are put in the physical form notes field.

It is noted whether the measurement is **actual** (*i.e.,* the entire length was available for measurement), **visible** (*i.e.,* part of the stone was not available for measurement because it was embedded in the ground or built into a wall), **uncertain** (it is impossible to tell from the published account), **inc**, **ind** or **n/a**. The last is used for unmeasured lost stones in particular.

The minimum visible measurement is fixed conventionally as 0.01m. If the information is not known it is given as 0.00m (because the field is numeric therefore **inc** is a not possible entry).

Following the precedent of the British Academy Corpus of Anglo-Saxon Stone Sculpture, and because most of the published corpora only provide measurements in imperial units, each dimension is expressed in both metric and imperial units (metric since it is the modern scientific standard, imperial since it is thought to approximate to the units in use in the early Medieval period). The source of each measurement is given (either CISP, or bibliographic reference). If measurements are available in only one format (metric or Imperial), the CISP data entry application provides automated conversion to the other system. In this case, the source is cited as **converted**.

Following the precedent of the British Academy Corpus of Anglo-Saxon stone sculpture, the imperial units used are feet and inches (conventionally expressed to 2 decimal places to allow for $\frac{1}{4}''$ *etc.*).

15. Source of data: **CISP**, reference, **n/a** or **converted**.
16. Height in metres.
17. **actual/visible/inc/n/aind/uncertain** (*i.e.,* source not clear on this point)
18. Width in metres.
19. As for 17.
20. Thickness in metres.
21. As for 17.
22. As for 15
23. Height, feet part.
24. Height, inches part.
25. As for 17.
26. Width, feet part.
27. Width, inches part.
28. As for 17.
29. Thickness, feet part.
30. Thickness, inches part.
31. As for 17.

32. Number of faces carved (excluding textual carving): **1 2 3 4 5 5+**
33. †Other carving notes: additional information on carving other than text (see also the INSCROSS and DECORATN tables as appropriate).

34. Has the stone been seen by CISP? Logical (yes/no) field.

3.3.2 The LOST table

Details of the date and circumstances of the loss of lost stones.

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
§2	`Stone`	numeric	–
3	`Date_last_present`	numeric	–
4	`Date_missing`	numeric	–
†5	`Notes`	memo	–

1. §Site code.
2. §Stone number.
3. Year last recorded present
4. Year first recorded missing
5. †Notes (authority, bibliographic reference, circumstances, places looked *etc.*)

Fields 1 and 2 provide a link to the stone table.

3.3.3 The OTHERLOC table

If stone is no longer in its original/earliest location (*i.e.,* the place described in 'site') and is not in a museum, its current location is given as a grid reference. See the GRID_REF table for more details on the details of the grid references.

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
§2	`Stone`	numeric	–
3	`Placename`	alphanumeric	35
†4	`Notes`	memo	–
5	`System`	alphanumeric	7
6	`East`	numeric	–
7	`North`	numeric	–
8	`Accuracy`	numeric	–

1. §The site code.
2. §The stone number.
3. The placename of the location.
4. †Any notes about this location.
5. The grid system (*i.e.,* **GB**, **Ireland**, **France**).
6. The easting as a six figure coordinate.
7. The northing as a six figure coordinate.
8. The true accuracy of the reference in metres.

3.3.4 The MUSEUM table

Information about any stone currently housed in a museum (but not a church).

Field	Field Name	Type	Length
§1	`Site`	alphanumeric 5	
§2	`Stone`	numeric	–
3	`Mus`	alphanumeric	30
4	`Acc_no`	alphanumeric	30

1. §The site code.
2. §The stone number.
3. Name of museum.
4. Museum's accession/catalogue number.

3.3.5 The FORM table

This is a hierarchical look-up table of allowed forms.

Field	Field Name	Type	Length
§1	`Mon_form`	alphanumeric	8
2	`Gen_form`	alphanumeric	25
3	`Med_form`	alphanumeric	25
4	`Detail_form`	alphanumeric	25

1. §The form code—a five character letter code in lowercase.
2. The general categories of monument.
3. The middle level of detail of category of monument.
4. Detailed classification of monument form.

The following is a listing of the FORM table and is then followed by definitions of some of the terms.

`Mon_form`	`Gen_form`	`Med_form`	`Detail_form`
arcar	architectural element	arch	arch
arcbs	architectural element	building stone	building stone
arcli	architectural element	lintel	lintel
arcot	architectural element	other	other
arcpa	architectural element	panel	panel
block	block	block	block
colum	column	column	column
eccal	ecclesiastical furniture	altar	altar
eccfn	ecclesiastical furniture	font	font
eccot	ecclesiastical furniture	other	other
fcbas	free-standing cross	cross-base	cross-base
fcsA	free-standing cross	cross-shaft	Cramp 1a
fcsA	free-standing cross	cross-shaft	Cramp 1b
fccA1a	free-standing cross	complete	Cramp sh. A, head 1a

Contents of the FORM table continued…

Mon_form	Gen_form	Med_form	Detail_form
fccB6e2	free-standing cross	complete	Cramp sh. B, head 6e, r2
fccB8e5	free-standing cross	complete	Cramp sh. B, head 8e, r5
fch1a	free-standing cross	cross-head	Cramp head 1a
etc.	*etc.*	*etc.*	*etc.*
fscuc	free-standing cross	otherwise unclassified	otherwise unclassified
fccgw	free-standing cross	complete	Glams. 'wheel headed' type
fragy	fragment	fragment	fragment
nboul	natural	boulder	boulder
ncave	natural	cave	cave
nrock	natural	rock outcrop	rock outcrop
pillc	pillar	slab (flat)	cross-marked
pillp	pillar	slab (flat)	plain
pilsc	pillar	stone (squarish)	cross-marked
pilsp	pillar	stone (squarish)	plain
recmo	recumbent monument	recumbent monument	recumbent monument
rpmeg	re-used	prehistoric	megalith
rpout	re-used	prehistoric	carved rock outcrop
rralt	re-used	Roman	altar
rrarc	re-used	Roman	architectural fragment
rrmil	re-used	Roman	milestone
rroth	re-used	Roman	other
sarco	sarcophagus	sarcophagus	sarcophagus
slbbd	worked slab	body-slab	body-slab
slbcr	worked slab	cross-slab	cross-slab
slbns	worked slab	name-slab	name-slab
slbot	worked-slab	other	other

This list of terms is *not* complete as not all the possible forms of cross, cross shaft and base taken from Cramp's (1984) typology were entered as necessary. Figures 3.2–3.4 show Cramps classification. The codes are created as follows

1. First three letters are **fcc** (free-standing cross, complete), **fcs** (free-standing cross, cross-shaft) or **fch** (free-standing cross, cross-head).
2. Append shaft type (**A** or **B**) if appropriate.
3. Append head-type from Cramp's table (see figure 3.3), *e.g.,* 6e.
4. Append number for ring-type (see figure 3.4), *e.g.,* 5.

Additionally, there is a separate category for 'Glamorganshire disk' or 'wheel headed' crosses (**fccgw**).

The following provides additional definitions of some of the terms encountered in the above table.

reused

prehistoric

megalith A menhir, dolmen or component of stone circle.

carved rock outcrop A cup-marked surface or similar.

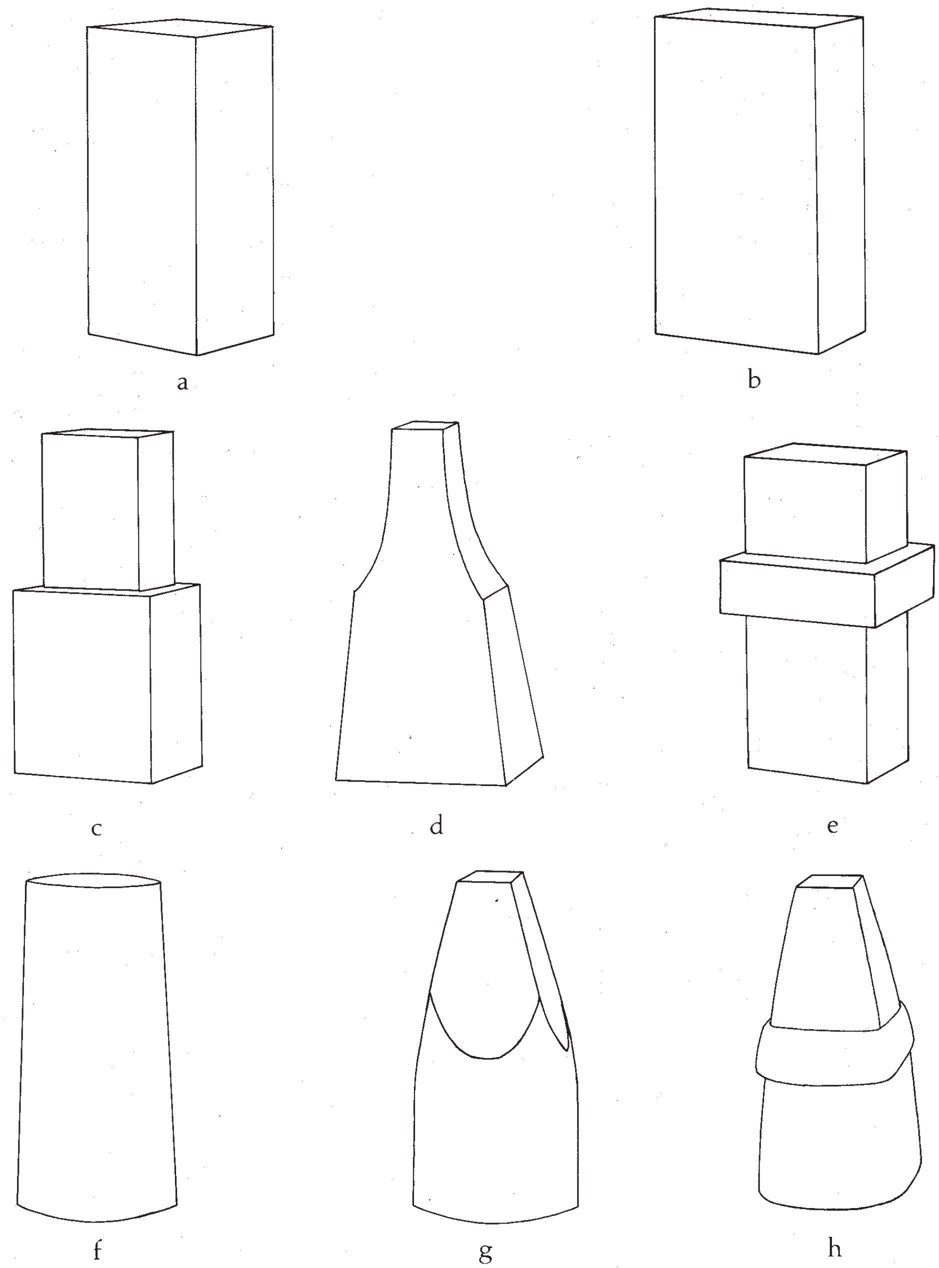


Figure 3.2: Cramp shaft forms

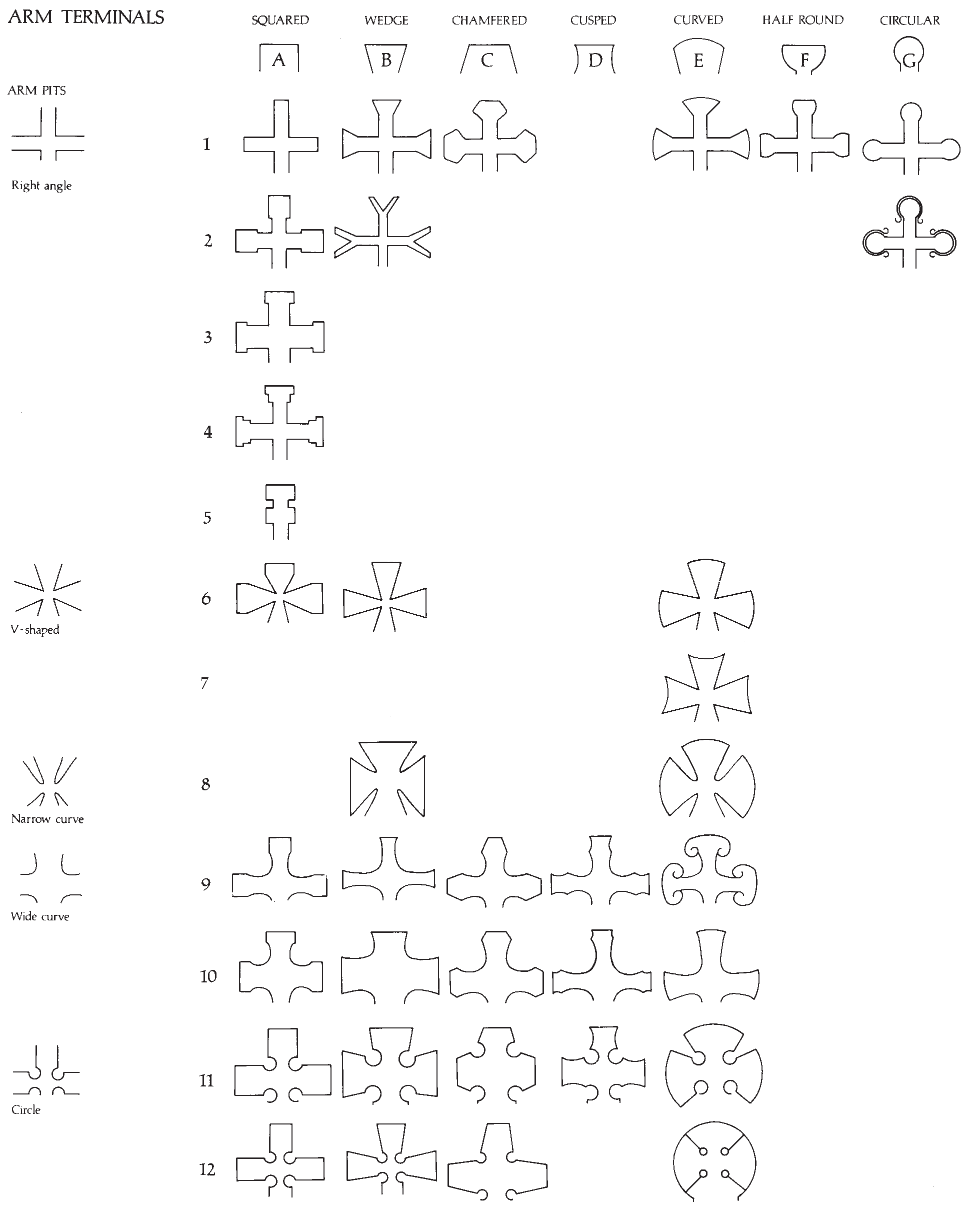


Figure 3.3: Cramp arm and terminal forms

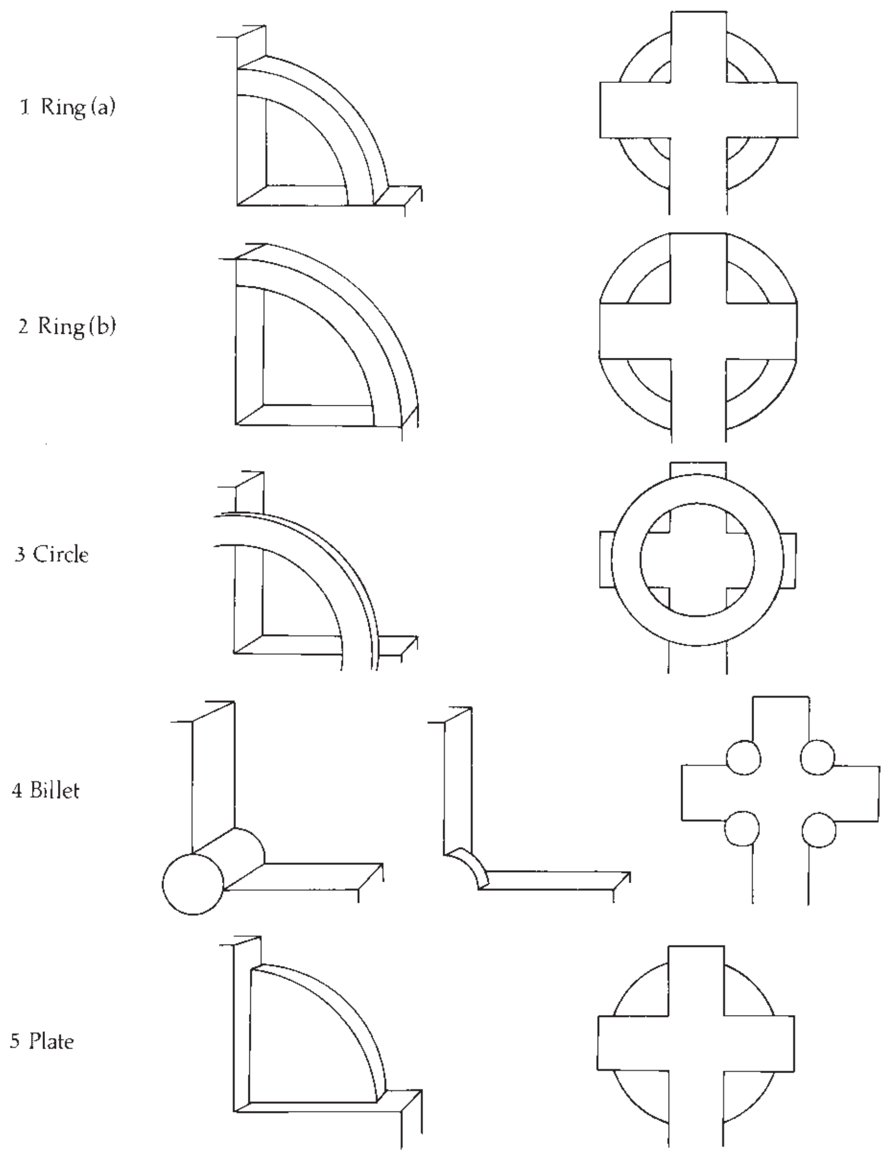


Figure 3.4: Cramp ring forms

natural

- **rock outcrop** As carved rock outcrop above but no earlier carving.
- **boulder** Boulder—differs from an outcrop in not being attached to ground, but not tall and thin like a pillar, and is unshaped.

architectural element

- **building stone** *e.g.,* undifferentiated wall block
- **lintel** of a door or window
- **panel** *i.e.,* a flat slab designed to be set into the wall of a building, *e.g.,* dedication slab, frieze
- **arch** (component)
- **other**

burial monument

- **sarcophagus** *i.e.,* hollow burial monument, with or without lid
- **recumbent** a thick body cover, *i.e.,* not a flat slab but carved on vertical sides as well as top

pillar-stone tall and thin, squarish or round cross-section, unworked or lightly worked

- **cross-marked** with cross
- **plain** no carving other than text

pillar-slab tall and thin, flat face, rectangular cross-section, unworked or lightly worked

- **cross-marked** with cross
- **plain** no carving other than text

column tall and thin, squarish or round cross-section, fully worked.

block thick squarish lump of worked stone not otherwise differentiated of which the function is not clear, not a fragment since substantially intact, not building stone since carved on more than one surface.

worked slab thin, rectangular or square and substantially worked

- **name-slab** these would usually be small — less than $4'$ long, are more likely to be squarish than elongated, tend to be recumbent — carved on one side only, and be carved with little ornament beyond a simple cross, the name predominates in the over all design (also known as a 'pillow stone' but this term avoided (a) to avoid predetermining function and (b) as this term is strongly associated with Northumbria, to use it might imply influence)
- **body-slab** like the above but larger, usually elongated, usually $5'$ or $6'$ long and 2–$3'$ wide; cross often more substantial and predominates. The difference between these and recumbent monuments is that body-slabs are thin and quasi-two dimensional, and are not decorated on the vertical faces.
- **cross-slab** upright, usually carved on at least two sides, usually having a cross dominating but often with geometric or figural carving
- **other** *e.g.,* uprights without cross

fragment *i.e.,* indeterminate — too small to identify

3.3.6 The ALTSNAME table

Alternative name for stone. A stone may have more than one.

For example, local traditional name for stone or any other name under which the stone has appeared in print; *e.g.,* the Newton Stone, referred to in earliest accounts as 'Pitmachie stone', and also the 'Shevock stone' after nearby places. The Camp Ogham stone known locally as 'Faisi's grave'.

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
§2	`Stone`	numeric	–
§3	`Alt_sname`	alphanumeric	30

1. §Site code
2. §Stone number
3. §The alternative name

3.3.7 The FOLKLORE table

Folklore associated with stone (disregard things pertaining to site only).

Field	Field Name	Type	Length
$ 1	Site	alphanumeric	5
$ 2	Stone	numeric	–
†3	Folklore	memo	–

1. The site code.
2. The stone number.
3. †Notes concerning any traditional beliefs or practices pertaining to the stone.

Since only a small minority of stones will have associated folklore it is noted in a separate table.

3.3.8 The INSCROSS table

This table records the form of the inscribed crosses which may occur on a stone.

Field	Field Name	Type	Length
§1	Site	alphanumeric	5
§2	Stone	numeric	–
§3	Cross	numeric	–
4	Shape	alphanumeric	10
5	Substance	alphanumeric	10
6	Arms	alphanumeric	10
7	Terminals	alphanumeric	10
8	Crossing	alphanumeric	10
9	Frame	alphanumeric	10
10	Ring	alphanumeric	10
11	Base	alphanumeric	10
12	Interior	alphanumeric	10
13	Chi_rho_hook	logical	–
14	Other_embellishment	logical	–

1. §The site code.
2. §The stone number.
3. §The cross number—a stone can have more than one cross.
4. Shape: **equal-armed**, **latin**, **arcs**, **tau**, **inc**, **ind**, **n/a**.
5. Substance: **linear**, **outline**, **interlace**, **inc**, **ind**, **n/a**.
6. Arms: **straight**, **tapering**, **expanded**, **inc**, **ind**, **n/a**.
7. Terminals: **plain**, **curved**, **round**, **half-round**, **bifid**, **expanded**, **crosslet**, **other**, **mixed**, **inc**, **ind**, **n/a**.
8. Crossing: **plain**, **square hollow**, **round hollow**, **curved**, **circular**, **square**, **lozenge**, **inc**, **ind**, **n/a**.
9. Ring: **none**, **inner curved**, **outer curved**, **billet**, **angular**, **inc**, **ind**, **n/a**.
10. Base: Any of the terminal types, plus **angular** and **tenon**.
11. Interior: **plain**, **decorated**, **inc**, **ind**, **n/a**.
12. Frame: **none**, **circular**, **angular**, **cruciform**, **inc**, **ind**, **n/a**.
13. Chi-Rho Hook: logical field, yes/no.
14. Other structural embellishment: logical field, yes/no.

Some of the terms used above are now further defined.

Substance

interlace *i.e.,* constructed out of interlace, not an outline cross decorated with interlace.

Base

tenon *i.e.,* some kind of point

Chi-Rho Hook proper chi-rho monograms are not crosses, this covers only crosses with a little 'hook' on the upper arm.

Other embellishment *i.e.,* some other structural aspect of the cross, *e.g.,* indentations in stem, scrolls or billets attached to stem — anything on the surrounding background will be covered in the DECORATN table.

Some types of cross inevitably have default entries in other fields. For example, a linear cross must have **n/a** for interior.

Figure 3.5 shows the meanings of the various terms in diagramatic form.

3.3.9 The DECORATN and DECOR_CD tables

As any one stone many have more than one type of decorative element, and any one decorative element can occur on more than one stone, we have a many-to-many relationship which has to be resolved by the use of the third linking table. The DECOR_CD table lists the possible decorative elements in a codified form, and the DECORATN table acts as a linking table between the STONE table and the DECOR_CD table. The DECOR_CD table is a look-up table similar to SITE_TYP.

The DECOR_CD table has the following structure.

Field	Field Name	Type	Length
§1	`Type`	alphanumeric	6
2	`Description`	alphanumeric	35

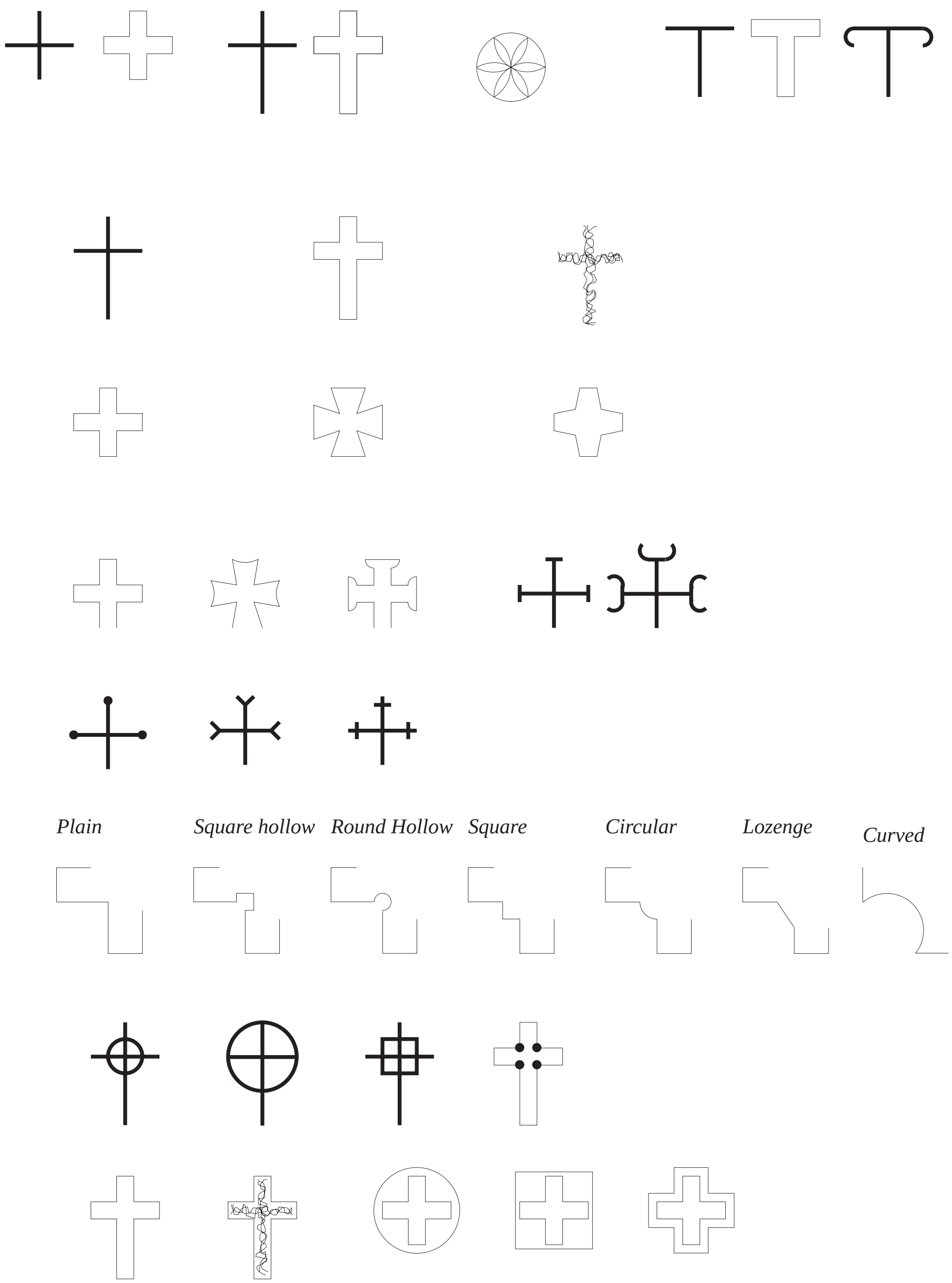


Figure 3.5: Guide to inscribed cross forms

The entries in this look-up table are listed below.

Type	Description
animal	animal
band	band
beadg	beading
boss	boss
chirho	chi-rho
figure	figural
frame	frame
geoinr	geometric ribbon interlace
geoinz	geometric zoomorphic interlace
geokey	geometric key pattern
geooth	geometric other
geospl	geometric spiral
inc	incomplete data
ind	indeterminate
other	other

The structure of the linking table DECORATN is as follows.

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
§2	`Stone`	numeric	–
§3	`Decor_code`	alphanumeric	6

1. §The site code.
2. §The stone number.
3. §The decoration code which is limited to one of the entries in the `type` field of the DECOR_CD table.

As can be seen, any one stone can have many entries in this table, as can any one decoration type. From this we can retrieve either all the elements on a stone, or all stones which employ that element.

3.4 The INSCRIP and related tables

Tables considered in this section are: INSCRIP, NAMES, SPEC_CHR, SPECIAL, and DATE. Related look-up tables not discussed in detail are DAMAGE, ORIENTAT, POSIT1, POSIT2, POSIT3, POSIT4, GENDER and LANGUAGE.

3.4.1 The INSCRIP table

This table is the primary table for recording details of the inscriptions *other* than the text itself which is recorded in the READING and TRANSLAT tables. A stone may have more than one inscription. The definition of a 'separate inscription' is a subjective decision and must be based on common sense, *e.g.,* if one obviously post-dates another, if they are in manifestly different 'hands', different scripts, on different faces of the stone, relate to different fields of ornament *etc.*

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
§2	`Stone`	numeric	–
§3	`Inscription`	numeric	–
4	`Doubtful`	logical	–
5	`Incomplete`	logical	–
6	`Condition`	alphanumeric	12
†7	`Legibility`	memo	–
8	`Secondary`	logical	–
9	`Orientation`	alphanumeric	4
10	`No_lines`	numeric	–
11	`Position1`	alphanumeric	5
12	`Position2`	alphanumeric	6
13	`Position3`	alphanumeric	20
14	`Position4`	alphanumeric	20
15	`Position`	memo	–
16	`Incision`	alphanumeric	10
17	`Technique`	memo	–
18	`Names`	numeric	–
19	`Language`	alphanumeric	5
†20	`Ling_notes`	memo	–
21	`Carve_err`	alphanumeric	1
22	`Script`	alphanumeric	5
23	`Letter_height_m`	numeric	–
†24	`Paleo_notes`	memo	–

1. §The site code.
2. §The stone number.
3. §Number assigned by CISP to act as primary key in conjunction with the site and stone fields. Number to be assigned 'within' a stone, not sequentially.
4. Is there some reason to doubt whether or not the inscription should be included? (*i.e.,* the marks are indistinct or fragmentary and might not be a genuine inscription; a description of a lost stone is vague; an inscription may be a fake, or may be modern). **N.B.** If a stone can be definitely excluded it should be, this is only if doubts remain. Logical yes/no field.
5. Is there any reason to think that the inscription is incomplete? Logical yes/no field.
6. What is the state of preservation? Allowed entries: **good**, **some-**damage, **poor**, **n/a**. Values stored in the DAMAGE table.
7. †Notes on the legibility of the text.
8. Is the inscription obviously secondary? This can either be the second, later inscription on a stone, or where a stone has been used for an inscription secondarily to its function, *e.g.,* a graffito. Logical yes/no field.

9. Orientation of text relative to vertical or long axis of stone (as stone originally stood, or, in the case of stone inverted and re-used, as it stood to bear this text). The allowed entries (stored in the ORIENTAT table) are:

circ	circular
horz	horizontal
inc	incomplete entry
ind	indeterminate
mix	mixed directions
othr	other
verd	vertical down
vind	vertical indeterminate
vruu	vertical up up
vua	vertical up along
vuad	vertical up along down
vup	vertical up
vupd	vertical up down

Some entries are only applicable to Ogham inscriptions, *e.g.,* **vuad**.

10. Number of lines of text. A continuous line of text which turns corners *etc.* is 1 line.

11. Position 1, text relative to compass points: applies only to stones in landscape or churchyard settings which are not clearly secondary, *i.e.,* pillar-stones, pillar-slabs, cross-slabs, free-standing crosses, recumbents. Allowed values stored in look-up table POSIT1.
    - For upright monuments — which face or arris is inscribed? If text extends over more than one, list the face/arris on which text begins.
    - For recumbents — in which direction does the top of the slab point?

    Allowed entries: **N S E W NE NW SE SW many all n/a ind inc** which are stored in look-up table POSIT1.

12. Position 2, text relative to monument:
    - cross-slabs — **broad**, **narrow**, **both** (broad and narrow)
    - free-standing crosses — **head**, **shaft**, **base**, **more** (than one)
    - pillars *etc.*— **arris**
    - others — **top**, **other**, **n/a**, **ind**, **inc**

    Allowed values stored in look-up table POSIT2.

13. Position 4, text relative to inscribed cross, if applicable: **above cross**; **below cross**; **beside cross**; **inc**; **ind**; **mixed**; **n/a**; **on cross**; **within quadrants**. Allowed values stored in table POSIT3.

14. Position 4, text relative to carving other than inscribed crosses. Allowed values stored in look-up table POSIT4. These are: **inc** (incomplete data); **ind** (indeterminate); **moulding** (on a raised band or moulding); **other** (all other possibilities); **panel** (enclosed on all sides by a panel); **quadrant** (within quadrant(s) of a cross); **separated** (divided from other ornament by lines); **undecorated** (undecorated); **undivided** (in same field as other ornament).

15. Incision — type of cut, *e.g.,* pocked, cut *etc.*. The controlled vocabulary of this field is yet to be determined.

16. Technique — memo field for the technique of letter carving noting, in particular, differences of opinion.
17. Number of individuals named (*i.e., X* son of **Y** is one person, not two).
18. Language of text other than personal-, place-, or ethnic-names. Allowed terms are stored in the look-up LANGUAGE and include: **AS** (Anglo-Saxon), **biblic** (Biblical), **briton** (Brittonic), **gaul** (Gaulish), **goidel** (Goidelic), **greek** (Greek), **inc** (incomplete data), **ind** (indeterminate), **latcel** (Latinised Celtic), **latin** (Latin), **mixed** (Celtic and Latin), **n/a** (not applicable), **name** (inscription consists solely of a name), **norse** (Norse), **r-celt** (Romano-Celtic), **unknwn** (unknown).
19. †Linguistic notes. Although, logically, these pertain to 'readings' rather than 'inscriptions', for practical reasons they are placed here. 'Linguistic' is loosely interpreted as any notes to do with the meaning of the inscription. See also the memo field in the NAMES table, and the linguistic indexing system.
20. Has the carver made an error in carving? (**N.B.** does not include textual 'errors'): **y**es, **n**o **c**orrected (*i.e.,* self-corrected), **i**nc.
21. Kind of script used: **rcaps** (predominantly Roman capitals), **rbook** (predominantly Roman book-hand), **ogham** (no-stem, *i.e.,* 'non-scholastic'), **oghms** (Ogham with-stem, *i.e.,* 'scholastic'), **runes**, **greek**.
22. Maximum letter-height at surface of incision in mm.
    - Roman alphabet — largest letter
    - ogham — largest H/B stroke
23. †Palaeographic notes: any points of interest concerning form of script used. This field is loosely interpreted and can be used for notes regarding the reading of the inscription too.

3.4.2 The NAMES table

Information concerning names in inscriptions. There can be more than one name in an inscription, each receives one row of data. Compound names receive multiple rows too, *i.e.,* Fred FILI Bert would have an entry for Fred and Bert. If authorities disagree on the reading of the name, a row can be input for each version of the name. Some mark-up as for the transcription of inscriptions is allowed, for example the use of square brackets (`[ ]`) or indications of missing text (*e.g.,* `[--`).

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
§2	`Stone`	numeric	–
§3	`Inscription`	numeric	–
§4	`Name_id`	numeric	–
5	`Name`	alphanumeric	30
6	`Language`	alphanumeric	10
7	`Gender`	alphanumeric	1
8	`Person_id`	logical	–
†9	`Name_notes`	memo	–

1. §The site code.
2. §The stone number.

3. §The inscription number.
4. §A sequential number to act as part of the primary key. This number has no meaning beyond being an identifier.
5. Text of name (as it appears on stone).
6. Language of name—as for `language` in the INSCRIP table (also uses look-up table LANGUAGE).
7. Gender of name: **m**ale, **f**emale, **e**ither, **u**nknown, **n**/a, **i**nc. Uses look-up table GENDER.
8. Known: can the person be identified from historical sources? Logical yes/no field. This is reserved for compelling identifications only.
9. †Any historical or linguistic notes on the name.

3.4.3 The DATE table

A published opinion on the date of an inscription. An inscription may have more than one of these.

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
§2	`Stone`	numeric	–
§3	`Inscription`	numeric	–
§4	`Date_id`	numeric	–
5	`Authority`	alphanumeric	25
6	`Date_from`	numeric	–
7	`Date_to`	numeric	–
8	`Date_notes`	memo	–

1. §The site code.
2. §The stone number.
3. §The inscription number.
4. §A number assigned by CISP.
5. Source of date (links to `bib_ref` field of the BIBLIOG table.
6. Start of date range.
7. End of date range.
8. Notes concerning the date, usually a quotation from the authority concerned.

Many of the dates are ‘conventional’ and are entered in the following manner:

Phrase	Date from	Date to
6th century	500	599
6th to 7th centuries	500	699
mid 6th to mid 7th centuries	550	650
late 6th to early 8th centuries	566	733

This conventional phrases will eventually be stored in a look-up table to enable easier extraction of inscriptions of certain dates. More precisely dated inscriptions, *i.e.,* those that mention a specific person, will use those dates.

3.4.4 The SPEC_CHR and SPECIAL tables

The structure and use of these two tables is similar to those for carved decoration discussed in section 3.3.9 on page 36.

These tables enable the tagging of distinctive letter-forms and other palaeographical features. As an inscription may have more than one special character, and a special character can be found in more than one inscription, this is a many-to-many relationship which is resolved through the use of a linking table. In this case, table SPECIAL is the linking table and SPEC_CHR is the list of special characters. The latter can also been seen as a look-up table.

The structure of the SPECIAL table is:

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
§2	`Stone`	numeric	–
§3	`Inscription`	numeric	–
§4	`Spec_char`	alphanumeric	5

1. §The site code.
2. §The stone number.
3. §The inscription number.
4. §The special character code which links to the SPEC_CHR table.

The structure of the SPEC_CHR table is:

Field	Field Name	Type	Length
§1	`Spec_char`	alphanumeric	5
2	`Character`	alphanumeric	40

1. §The character code and primary key.
2. Explanation or full name.

As data entry progresses it will become clear which characters need to be flagged. The following are examples only

- roman predominantly capitals
  - horizontal-I
  - angle-bar-A
  - retrograde-S
  - retrograde-N
- roman predominantly bookhand
  - punctuation
  - initial cross
  - dividing cross
  - suspension mark
- ogham
  - the various supplementary letters
  - bound letters
  - word-division

– directional indicator

These tables will eventually be expanded into a palaeographic indexing system.

3.5 The READING table

The only other table directly related to READING is the linking table READ_PUB which will be discussed in section 3.7.

This table contains published readings of text (or CISP reading if visited). An inscription may have many of these.

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
§2	`Stone`	numeric	–
§3	`Inscription`	numeric	–
§4	`Reading`	numeric	–
5	`By_whom`	alphanumeric	20
6	`When`	numeric	–
7	`Text`	memo	–

1. §The site code.
2. §The stone number.
3. §The inscription number.
4. §The reading number assigned by CISP. All four fields *together* act as the primary key.
5. Name of authority in full, *e.g.,* Macalister, R. A. S.
6. Date of this reading noted, because the authority may give different readings at different times and the latest is not always the best, *e.g.,* if it is many years since the stone was actually seen and the reading has been 'tidied'; stone may have deteriorated physically over time, *etc.*. If not date for reading given, date of publication used. If multiple visits cited, date of last visit.
7. Text of reading, *i.e.,* transliteration of inscription—see below.

Transliteration of inscriptions

Obviously, established conventions should be followed wherever possible, however:

1. Although there is substantial convergence among the various systems there are numerous difference in detail.
2. Existing systems draw heavily on typographical features which are unavailable to us in the database (*e.g.,* subscript dots) or are likely to prove too cumbersome to achieve (*e.g.,* italics).
3. Existing systems use new lines of printed text to reflect layout of inscribed text whereas we must input ours as one continuous string of data.

As a result, therefore, some innovation is unavoidable.

As we see it, the main drawbacks for our purposes with existing systems are that they distinguish between too many levels of uncertainty, and that they represent transliteration and expansion of abbreviations at a single stage. By separating out these last two functions we need display less information at each stage.

As there are unrestricted notes on completeness and legibility of the text in other fields, what is needed is a simple, comprehensive and accurate transcription of what is actually on the stone. Abbreviations are expanded (and lacuna supplied) in a separate table.

In what follows CISP innovations are marked with a #

All inscriptions

The prevailing convention is to transcribe as upper-case, regardless of whether or not the originals are capitals (runic inscriptions are conventionally transcribed as lower case).

	A space in the transliteration represents a deliberate space in the inscription (except either side of a `|`).
`|`	End of line.
`||`	Text interrupted by a zone of ornament.
`|||`	# Text turns corner (not indicated as such in other systems).
[	] # Material contained in square brackets [] is defective or difficult to read (damaged or missing) and supplied by the reader.
{ }	# Material in curly brackets {} is clear on stone but is in some way unusual (*e.g.,* an inverted letter) or, though legible, is difficult to transliterate because the value is unknown or disputed (*e.g.,* one of the more unusual of the supplementary Ogham letters) Defective letters which can be restored with certainty — Cramp (1984) and Okasha (1993) put these in square brackets. *RIB* does not indicate them if certain claiming that illustration shows clearly. CISP will follow *RIB* and only mark if a letter is certain, or uncertain. Doubtful letters — Okasha (1993) and Cramp (1984) indicate the preferred restoration of these with italics within square brackets — [*AB*], *RIB* uses a subscript dot. Neither of these options is available to us typographically. If we are not indicating the restoration of certain letters, then we use [AB]. # If there are two possible options then these can be indicated by `[^]`, *e.g.,* `A[B^C]D` represents `ABD` or `ACD`, `V[W^XY]Z` represents `VWZ` or `VXYZ`. The preferred option going first. This convention is intended primarily for use with Ogham inscriptions where there is often more than one option. In the interests of clarity only one alternative will be listed. If there is more doubt than this the character will be transcribed as ‘illegible’.
. . .	Illegible (number of letters known): follow Okasha/Corpus — `[..]` where one dot represents one letter.
– –	Illegible (number of letters unknown): follow Okasha/Corpus — `[--]` Lacuna of unknown length at beginning or end of a line: *RIB*, Okasha and Corpus all use different conventions. # CISP will use `--]` and `[--` which is consistent with above convention.

`/`	Conjoined or ligatured letters, *e.g.,* Æ: follow Okasha/Corpus and use `A/E`
`:`	Punctuation of any sort.
`+`	Textual crosses in the body of the inscription (*i.e.,* as punctuation, not decoration).
·	# Suspension marks. The tilde will follow each letter.
<>	# Character inserted by carver *e.g.,* `<AB>`
{}	# Special characters, *e.g.,* inverted characters, compendia (per, pro), otherwise anomolous characters (described in full in palaeographic notes field); *e.g.,* `{A}` a special form of A (angle-bar A, an inverted A, *etc.*), `{I}`, `{P}`
{*}	Character legible but either not identifiable, or not transcribable, *e.g.,* Ω
	# V for pointed character, U for rounded or square-bottomed character, regardless of whether consonantal or vocalic. *RIB* distinguishes in a similar way (though their material is different).

Ogham

All of the above conventions which are relevant apply. The following transliteration scheme is used:

BLVSN
HDTCQ
MGGwStR (Gw not Ng, St not Z — N.B. these are extremely rare epigraphically)
AOUEI
X represents first forfid (supplementary character — reasonably common)
If a value for a rare forfid is generally agreed, then transcribe as a special form of the letter, *e.g.,* angle-vowel A = `{A}` *etc.*
If a carved letter can be ascribed to a group but not specified more closely then use `!`, *e.g.,* `[B!]` represents a B or a subsequent member the B-group, `[T!]` represents T, C or Q. *etc.*.

3.6 The TRANSLAT table

This table records the expansion and interpretation of each reading.

Field	Field Name	Type	Length
§1	`Site`	alphanumeric	5
§2	`Stone`	numeric	–
§3	`Inscription`	numeric	–
§4	`Reading`	numeric	–
§5	`Expand_id`	numeric	–
6	`Expansion`	memo	–
7	`Translation`	memo	–

1. §The site code.
2. §The stone number.

3. §The inscription number.
4. §The reading number.
5. §The expansion number, assigned by CISP. All five fields *together* act as the primary key.
6. Expanded text of inscription (expanding contractions, suspensions or other abbreviations) Both this and the following field can use elements of the transliteration scheme from above where appropriate, in particular `[ ]` to represent uncertain letters and `--` or `...` to represent missing letters or parts of the inscription.
7. Translation/interpretation of above. Personal names are followed by `(PN)`. If no translation given in the published source, use **n/a**.

3.7 The bibliography subsystem

This section contains the primary BIBLIOG table, and a number of linking tables: SITE_PUB, STON_PUB, INSC_PUB, READ_PUB, CORPORA and IMG_PUB. Logically, there could also be a TRAN_PUB to link table TRANSLAT to BIBLIOG but in practice the entries would be identical to READ_PUB.

3.7.1 The BIBLIOG table

Full bibliographic references in a modified 'Chicago A' style will be stored. This table contains all bibliographic references. As a general principle only 'useful items' will be cited in bibliographies:

- include first mention
- include references to standard corpora or by major authorities (*e.g.,* Macalister, Rhŷs)
- include any item containing original information or substantive discussion
- include references with a good illustration
- exclude derivative or otherwise secondary discussion
- exlude mere mentions unless they come under one of the above headings

Unpublished 'official' documentation, *e.g.,* OS record cards in NMRs are to be cited where information is easily available.

The structure of the table is as follows:

Field	Field Name	Type	Length
§1	`Bib_ref`	alphanumeric	30
2	`Reference`	alphanumeric	255
3	`Date`	numeric	–
4	`Verified`	logical	–
5	`CISP_notes`	alphanumeric	40

1. §A primary key (code). This will be constructed in a manner similar to author-date reference systems, *i.e.,* name+year. Examples:
   — Lockyear 1996 — `Lockyear/1996`

— Reece 1987a, 1987b — `Reece/1987a` *and* `Reece/1987b` *i.e.,*, two separate records.
— Lockyear and Wilcock 1985 — `Lockyear/Wilcock/1985`
— Lockyear, Ponting and Poenaru Bordea 1997 — `Lockyear/etal/1997`.

2. Full bibliographic reference (see examples below).
3. Date of publication (this information, contained in the above field also, is duplicated here to permit searching by date, *e.g.,* all post CIIC refs. to a particular stone).
4. Has the reference been checked by CISP? Logical yes/no field.
5. For project use only to help in the checking process, locating references, *etc.*.

3.7.2 Sample bibliographic entries

There follow some example references in the modified form of 'Chicago A' in the form that they will be entered into the `ref` field of table BIBLIOG.

Book: `Thomas, C. (1993) _And Shall These Mute Stones Speak?_ Cardiff: University of Wales Press.`
Article:`Westwood, J. O. (1855) 'Notices of several Early Inscribed Stones recently found in various parts of Wales', _Archaeologia Cambrensis_ 3rd series, 1, 4--10.`
Article in volume:`Craig, D. (1991) 'Pre-Norman sculpture in Galloway: some territorial implications', _in_ R. D. Oram and G. P. Stell (eds) _Galloway Land and Lordship_, 45--62. Edinburgh: Scottish Society for Northern Studies.`
Two authors:`Dark, K. R. and S. P. Dark (1996) 'New Archaeological and Palynological...`
Three or more authors:`Daire, M.-Y., A. Villard, S. Hinguant and E. Le Goff (1996) 'Les steles de l'age du fer a decors geometriques...`
Editors: Use (ed.) for singular, (eds) for plural.

3.7.3 The SITE_PUB table

This table provides the link between a site, and the bibliographic information. This linking table also contains extra information regarding the connection between the two, *i.e.,* specific pages and value.

Field	Field Name	Type	Length
§1	Site	alphanumeric	5
§2	Bib_ref	alphanumeric	30
3	Pages	alphanumeric	40
4	Photo	alphanumeric	10
5	Drawing	alphanumeric	10
6	Map	alphanumeric	10
7	value	alphanumeric	25

1. §The site code which links to the SITE table.
2. §The bibliographic reference code which links to the BIBLIOG table.
3. Relevant pages, figure and table references.
4. Is there a photograph? Allowed entries **yes**, **no**, **inc**.
5. Is there a drawing. Entries as for photo.
6. Is there a map? Entries as for photo.
7. Value of the reference—by definition all references cited are non-trivial. All references are described with one of the following terms which are stored in the look-up table PUB_VALS:
   cdisc concise discussion (the most common entry)
   illus illustration of use only
   inc incomplete data
   list listing (of stones, inscriptions, or other people's readings)
   minor minor reference
   other other reference
   read reading only (for READ_PUB table only)
   sdisc substantial discussion
   N.B. The value of a given reference is relative to this specific site. *e.g.,* a single reference could be a substantial discussion of one site *but* contain only a minor reference to another site. The bibliography will be as comprehensive as possible and we would aim to give all substantial discussions and first mentions, but minor references and illustration only references will be included on an *ad hoc* basis.

3.7.4 The STON_PUB, INSC_PUB, and READ_PUB tables

These tables are identical to the SITE_PUB table discussed in the previous section with the exception of additional key fields providing the link to the primary tables, *i.e.,* the STON_PUB table has a `stone` field which, in conjunction with the `site` field, provide the link to the STONE table; the INSC_PUB table has `stone` and `inscription` fields, and so on. The INSC_PUB table only has `photo` and `drawing` illustration fields; the READ_PUB table has no illustration fields.

The conventions for `value` and `cisp_notes` fields are as for the SITE_PUB table.

3.7.5 The CORPORA table

This is a special version of the linking table between the INSCRIP table and the BIBLIOG table allowing retrieval of inscriptions via standard corpus references *e.g.,* CIIC, Nash-Williams, Okasha, Thomas *Mute Stones*, Kermode, and others as appropriate.

Field	Field Name	Type	Length
§1	Bib_ref	alphanumeric	30
2	Corp_no	alphanumeric	10
§3	Site	alphanumeric	5
§4	Stone	numeric	–
§5	Inscription	numeric	–

1. §The bibliographic reference code.
2. Designation in corpus.
3. §The site code.
4. §The stone number.
5. §The inscription number.

3.7.6 The NAME_BIB table

This provides bibliographic references for names stored in the NAME table.

Field	Field Name	Type	Length
§1	Site	alphanumeric	5
§2	Stone	numeric	–
§3	Inscription	numeric	–
§4	Name_id	numeric	–
§5	Bib_ref	alphanumeric	30
6	Pages	alphanumeric	30

1. §The site code.
2. §The stone number.
3. §The inscription number.
4. §The name number.
5. §The bibliographic reference code.
6. Relevant pages.

3.8 The image subsystem

The IMAGE table will contained scanned images of sites, stone and inscriptions.

The image table has the following structure:

Field	Field Name	Type	Length
§1	`Image_no`	numeric	–
2	`Type`	alphanumeric	10
3	`Label`	alphanumeric	60
4	`image`	graphic	–

1. §Sequential reference number.
2. Type of image: **b/w** (photograph) **colour** (photograph), **drawing**, **rubbing**, **squeeze**, **other**.
3. Label (*e.g.,* ‘View of site from SW’, ‘inscribed panel with letters chalked in’, ‘cast in National Museums of Scotland’).
4. The image itself.

This table can be linked to other primary tables via linking tables as in other cases above.

3.9 The linguistic indexing system

This system replaces the ‘formula’ fields/tables of the earlier version of the database.

The system consists of two tables. The first, the WORDINDX table .... COMPLETE

Appendix A

Glossary

This glossary contains explanations of some of the terms used in this manual. Cross-references to other entries are given in **bold type**.

application, database A database application is a computer program to help users of a specific database. **RDMS**s provide an *application* language so that a programmer can create this program.

attribute A piece of information about an item or **entity**. For example, hair colour is an attribute of a person, population is an attribute of a town.

column A database **table** will usually consist of many **rows** and columns, just like a table in a book. Each column will store information about an **attribute**. A table of people may have columns for height, hair colour, eye colour, *etc.*

database A structured collection of data.

database application See application, database.

database management system A computer program for storing and manipulating data.

DMS See database management system.

entity An group of items about which data is stored; for example 'stone' or 'site'. Entities are usually represented by a table or relation with one or more **attributes**; an entry in the table, *e.g.,* a particular stone, is an **instance**.

form A form in a computing context is a structured way of displaying and/or entering data on screen. It is analogous to the more common paper form.

instance A particular example or entry of an **entity**; *e.g.,* Welwyn Garden City is an instance of the entity 'towns'.

normalisation

relation A special type of table in which (a) every row is unique, (b) the columns are in no fixed order and (c) the rows are in no fixed order. All the tables in a **relational database** will be relations.

relational database is a particular way of structuring data which will consist of multiple **relations** and is manipulated using a **relational database management system**.

relational database management system A computer program for storing and manipulating data which has been structured to a form a **relational database**.

RDMS See relational database management system.

row A database **table** will usually consist of many rows and **columns**, just like a table in a book. Each row will store information about an **entity**. A table of people may have a row for each person.

table A set of data organised in a tabular form with **rows** and **columns** similar to a table in a book. A database usually consists of several tables. Tables many be linked via a **key field**.

tuple Another name for a **row**.

variable A term often loosely used to mean **attribute**. The term should be avoided as it has other meanings in a computing context.

Appendix B

Changes in the CISP database since original release of the manual

This appendix lists some of the more major changed in the database since the release of the first manual dated March 25th, 1997.

Logical fields

On the advice of Nick Ryan who kindly looked through the original manual, many logical fields were deleted. This is because they could create inconsistencies in the database. For example, it would have been possible to select 'Yes' for the logical field `Inscribed cross` without actually entering data for the cross in the appropriate table. By removing this field, one is forced to check for the presence of an inscribed cross by seeing if there is an entry in the appropriate table.

The SITE_TYP table

This table was simplified greatly from the original scheme as it was felt that in many cases it was difficult or impossible to assign sites to one of the original categories.

The DECOR_CD table

This table was simplified greatly from its original hierarchical form.

The archive subsystem

This was abandoned as this information did not require cataloging as part of this database. The primary part of the archive which will be included are the images which will be stored as part of the image sub-system.

Formulae and the linguistic index

The formula structure originally used was found to be unworkable in practice. This has been replaced by the linguistic indexing system.

Appendix C

Future work

Then following tasks need to be completed.

1. Images — images need to be scanned and then input to the database.
2. Linguistic indexing system needs to be completed and data input.
3. Expansion of the SPECIAL tables to create a palaeographic indexing system.
4. The creation of a Users database application on the Web or on CD. This must include:
   (a) Easy browsing and searching including images.
   (b) Possibility of printing reports.
5. The decoration codes and inscribed crosses need to be checked by someone with a greater knowledge of medieval art.
6. The monument form entries need to be checked for consistency.
7. The linking table between NAMES and the bibliography has not been filled in.
8. The REGION table is currently empty.
9. `site_history` and `site_description` were in practice difficult to split and in most cases all data has been entered to one field. This needs to be standardised by merging all entries to one field and dropping the other.
10. The `incision` field needs a controlled vocabulary, and entries standardised to it.
11. The grid references for all but the Dumnonia entries are lacking.

Bibliography

CARTER, J. 1992. *Programming in SQL*. Blackwell, Oxford.

CRAMP, R. 1984. *Corpus of Anglo-Saxon Stone Sculpture. Volume I, Part 1. County Durham and Northumberland*. Oxford University Press, Oxford.

CRAWFORD, M. H. 1969. *Roman Republican Coin Hoards*. Royal Numismatic Society, London. Special Publication No 4.

DATE, C. J. 1995. *An Introduction to Database Systems*. Addison-Wesley, Reading, Massachussetts, sixth edition. Inst Arch AK 20 Dat. 11

HENRIK, J. H. 1993. 'European archaeological databases: problems and prospects.' *In* J. Andresen, T. Madsen & I. Scollar (eds.), *Computer Applications and Quantitative Methods in Archaeology 1992*, pp. 229–237. Aarhus University Press, Aarhus. 4

LOCKYEAR, K. 1996. *Multivariate Money. A statistical analysis of Roman Republican coin hoards with special reference to material from Romania*. Ph.D. thesis, Institute of Archaeology, University College London. 4

MACALISTER, R. A. S. 1945. *Corpus Inscriptionum Insularum Celticarum*, volume 1. Stationary Office, Dublin.

MACALISTER, R. A. S. 1949. *Corpus Inscriptionum Insularum Celticarum*, volume 2. Stationary Office, Dublin.

OKASHA, E. 1993. *Corpus of Early Christian Inscribed Stones of South-West Britain*. Leicester University Press, London and New York.

Index