TGCW36
The New BNC Database
A Quick-and-Dirty Overview

Dominic Dunlop

Draft of 1st October, 1992

Contents

1 Introduction

During September, I developed a new database for the British National Corpus. It is now up, running, and (as I write) loaded with basic information on all texts received from OUP to date, and with information about texts being processed (that is, at the A_ and B_ stages, as described in TGCW35) as of 24th September.

This note describes the structure of the database, reports on its current status, sketches its relationship to the text header, and gives a few clues as to the sort of query that might produce useful results. A later, and more polished and complete, document will describe everything more fully, and give details of a worthwhile user interface, which has yet to be developed.

Please accept my apologies for the nastiness of the layout of this document, which results from the verbatim inclusion of the SQL used in the creation of the database tables. Presentation should be improved in a later version.

2 What's It For?

The BNC database is a project-wide resource which fills several rôles:

- Tracking the progress of texts through the "sausage machine";
- Generating throughput figures for use in project management;
- Recording bibliographic and other information used in the generation of CDIF headers for individual texts and for the corpus as a whole;
- Keeping track of the selection criteria applying to individual texts, generating aggregate figures which can be used in maintaining the balance criteria specified in the corpus design;
- Recording details of the permissions grantor (or grantors) associated with each corpus text;
- Giving corpus users a means of browsing for texts which satisfy their requirements. (This is a longer-term aim, which has not specifically been addressed at this stage.)

Initially, the database will be maintained solely by OUCS — although any consortium partner who wishes to query it may do so (subject to the setting-up of suitable communications links). It may eventually prove useful for other parties to enter information into the database, but we'll cross that bridge if we come to it.

3 Database Structure

For better or for worse, the database is based on the requirements of the text headers, as described in TGCW34, rather than on the original database design of TGCW05. It was my (somewhat hurried) judgement that so much had changed in terms of the requirements of the TEI header, our knowledge of the use of selection criteria, experience with obtaining permissions, and so on, that it would be easier to start from scratch, rather than to build on the existing database design.

3.1 Essential Information

There are probably four essential things that you need to know in order to use the database:

1. The most important table is `bncText`.
2. The primary key for `bncText` is `teKey`, which contains the six-character mixed-case unique filename for a given text.
3. Where information is not held in `bncText`, usually because it may occur one or more times for each text, or because it varies as a text progresses through processing, it is held in tables which use the value of `teKey` as a foreign key.
4. There are two types of table in the database: those which contain information about specific texts, and those used to look up and translate the codes used in the text information tables. You cannot tell from a table's name which category it falls into. (Sorry about that.)

In addition to these, for the moment, you probably need to know `SQL` in order to get much out of the database — although default *Ingres* forms can be used for simple browsing of individual tables.

If you want to know more than this, read the subsections that follow. These consist mainly of massaged tracts of the `SQL` source code used in the creation of the database tables. But they are well-commented.

3.2 Naming conventions

In order that one can tell what table a column belongs to, and what other column a foreign key column joins to, I adopted a set of conventions in generating names for database objects:

1. Only digits and upper- and lower-case letters are used in object names. Case is not significant in determining the uniqueness of an object name. (Indeed, *Ingres* maps all object names to lower case.)
2. "Hungarian" conventions are used to abbreviate multiple words. Thus, for example, "received from" becomes `RecFrom`.
3. All tables have names of the form `bncXyz...`, where `Xyz...` is some unique name.
4. All fields in table `bncXyz...` have names of the form `xyAbc...`
5. The primary key to a table `bncXyx...`, or the field in that table which, used in conjunction with a foreign key in the table, creates a unique primary key, is named `xyKey`.
6. A foreign key in table `bncPqr` whose main function is to join the field `xyAbc...` in table `bncXyz...` has a name of the form `pqXyAbc...`
7. Indexes for the table `bncXyx...` have names `xyIndex1`, `xyIndex2` ...

3.3 The bncText table

bncText is the most important table in the database, and contains items appearing at most once for each text submitted for inclusion in the BNC. Among other things, it forms the basis of the structured citation (if any) which appears in the text header source statement. Any further structured citations necessary for a full description of the source are handled by bncCite records (see below.)

The key, teKey, is unique. Indeed, it is unique independent of case. This restriction is enforced by the table bncUnique — see below.

The fields teSdKey, teChDate and teChWho should be automatically maintained. Currently, they are not.

```
create table bncText
(
    teKey           char(6)           not null, /* Unique name          */
    teRecName       varchar(32)       not null, /* Name of received file*/
    teRecFrom       char(8)           not null, /* From whom received   */
                                                /*   (Longman, OUP...)  */
    teRecDate       date              not null, /* Date received        */
    teSubDir        varchar(32),                /* Type assigned when   */
                                                /*   received (text,    */
                                                /*   newspapers etc.)   */
    teRecWords      integer           not null, /* Words when received  */
    teRecWho        char(8)           not null, /* Person receiving     */
                                                /*   (login name)       */

    /* There are four text types: WBP (written books & periodicals)     */
    /*                            WMI (written miscellaneous)           */
    /*                            SDE (spoken demographic)              */
    /*                            SCG (spoken context governed)         */
    /* Given this, one can work out what selection and classification   */
    /* features are expected, and whether there may be recording and    */
    /* script information.                                              */

    teType          char(3)           not null, /* Text type            */

    /* The following is the title of the monographic source text.  It   */
    /* should be entered exactly as it appears in the source text.      */
    /* Leading definite or indefinite articles should be left in place, */
    /* not deleted or moved.  (``The Need to Give'' is entered as such.)*/
    /* Where a name has to be generated (for example for collections or */
    /* spoken texts) this name should be enclosed in square brackets.   */
    /* Such ornaments as `` : an electronic sample'' should not be      */
    /* included: they get generated automatically.                      */

    teTitle         varchar(128)      not null, /* Title of source text */
    teGenTit        char(1)           not null, /* Generated title?     */
                                                /*   Y/N/U (unknown)    */

    /* Note that the imprint for all demographic and some selective     */
    /* spoken corpus texts is the BNC itself.                           */

    teImKey         char(6),                    /* Imprint (if any) for */
                                                /*   monographic text   */
    /* The following field has value ``Y'' if world-wide permissions    */
    /* controlled by agency named as imprint; ``N'' if permissions are  */
    /* known to be controlled by one or more agencies in addition to or */
    /* instead of the imprint agency; ``U'' if permissions status is    */
    /* unknown.  If the value is ``N'', there should be records in      */
    /* bncPermissions joining this record.  Field is null if teImKey is */
    /* null.                                                            */

    teImprintPerm   char(1)           not null, /* Perms from imprint?  */

    teISBN          char(10),                   /* ISBN (if any).  No   */
                                                /*   spaces, minuses.   */
    teSeKey         char(6),                    /* Serial (if any) of   */
                                                /*   which this is part */

    /* There are seven possible formats for the date.  All follow       */
    /* ISO 8601:1988 as far as possible.                                */
    /*              1991-12-31              Exact date                  */
    /*              1991-12                 Year and month              */
    /*              1991                    Year                        */
    /*              1991-12-01/1991-12-31   Range of days               */
    /*              1991-06/1991-12         Range of months             */
    /*              1989/1991               Range of years              */
    /*              ~1991                   Approximate year            */
    /* Where a range is given, associated analytic texts might be       */
    /* expected to have exact dates within the range.                   */

    teDate          varchar(21)       not null, /* Publication/         */
                                                /*   production date    */

    /* The following field duplicates information held in the bncStatus */
    /* table.  It has the following values:                             */
    /*              .   Newly received                                   */
    /*              -   Bounced (will not progress unless resubmitted)   */
    /*              A   Undergoing syntactic and semantic check          */
    /*              B   Queued for transmission to Lancaster             */
    /*              C   Received from Lancaster                          */
    /*              D   Undergoing final check                           */
    /*              E   Queued for corpus accession                      */

    teSdKey         char(1)           not null, /* Current status       */

    teChDate        date              not null, /* Date of last change  */
                                                /*   to information for */
                                                /*   this text          */
    teChWho         char(8)           not null  /* Changer (login name) */
);
create unique index teIndex1 on bncText(teKey);
```

3.4 Text data tables

3.4.1 `bncAnalytic`

This is a table containing one record for each analytic work in the corpus. The information goes into the header source description, the script statement or the recording statement, depending on the value of `anCiKey`. Analytic texts within a monographic text are numbered serially by `anKey`, starting from one.

Information from this table appears in structured citations in the BNC header source, script, or broadcast statements.

The database currently contains no information about analytic texts. It will need eventually to be generated, hopefully automatically, in order that we can enforce the "no more than 120,000 words from any one author" selection rule. Note that, unlike monographic texts, which have (potentially different) word counts recorded in `bncStatus` at each stage of their progress through the "sausage machine", analytic texts have word counts recorded once and for all here — probably at a late stage in processing.

```
create table bncAnalytic
(
    anTeKey         char(6)           not null, /* Six character code   */
    anCiKey         char(1)           not null, /* Type of citation     */
    anKey           smallint          not null, /* Number of text       */

    /* See notes relating to teTitle                                    */

    anTitle         varchar(128)      not null, /* Analytic text title  */

    /* There are four possible formats for the date.  All follow        */
    /* ISO 8601:1988 as far as possible.                                */
    /*              1991-12-31              Exact date                  */
    /*              1991-12                 Year and month              */
    /*              1991                    Year                        */
    /*              ~1991                   Approximate year            */

    anDate          varchar(21)       not null, /* Publication/         */
                                                /*   production date    */
    anImKey         char(6),                    /* Keys an imprint      */

    /* The following field has value ``Y'' if world-wide permissions    */
    /* controlled by agency named as imprint; ``N'' if permissions are  */
    /* known to be controlled by one or more agencies in addition to or */
    /* instead of the imprint agency; ``U'' if permissions status is    */
    /* unknown.  If the value is ``N'', there should be records in      */
    /* bncPermission joining this record.   Field is null if anImKey is */
    /* null.                                                            */

    anImprintPerm   char(1),                    /* Perms from imprint?  */

    anWords         integer                     /* Words in analytic    */
                                                /*   text               */
);
create index anIndex1 on bncAnalytic(anTeKey);
create unique index anIndex2 on bncAnalytic(anTeKey, anKey);
```

3.4.2 bncAuthEd

This table contains one record for each author or editor of each analytic, monographic or serial work represented in the corpus. This information goes into the header source description, the script statement or the recording statement, depending on the value of `auCiKey`. Applicability to analytic, monographic or serial texts is determined by reference to `auKey`.

Note that this information is not currently normalized: if an author or editor is associated with multiple works, they are named multiple times. We may or may not decide to normalize at some future time. Personal names are entered in the form `Lastname, First names or initials`; corporate names are entered verbatim.

There is currently no information about authors or editors in the database, other than that in the `bncOUPdb` table. This will be transferred semi-automatically to the `bncAuthEd` table in the near future.

```
create table bncAuthEd
(
    auTeKey          char(6)            not null, /* Six character code   */
    auCiKey          char(1)            not null, /* Type of citation     */

    /* Following field is zero where information relates to a           */
    /* monographic text; negative for a serial; positive for an analytic*/
    /* text.                                                             */

    auAnKey          smallint           not null, /* Analytic text no.    */

    auKey            integer1           not null, /* Key for ordering     */

    auAuthEd         char(1)            not null, /* Author/editor? (A/E) */
    auName           varchar(64)        not null, /* Name                 */
    auBirthYear      char(4)                      /* Year of birth (if    */
                                                  /*   known)             */
);
create index auIndex1 on bncAuthEd(auTeKey, auCiKey, auAnKey);
create unique index auIndex2 on bncAuthEd(auTeKey, auCiKey, auAnKey, auKey);
```

3.4.3 bncCite

This table contains one record for each additional structured citation required fully to describe a corpus text. There are two cases, enumerated by ciKey:

`B` Citation is for a broadcast statement

`S` Citation is for a script statement

Conceptually, there is a third case:

`W` Citation is for a written text source description

Because structured citations for written texts are generated from information in the corresponding bncText record, no records with a `ciKey` of `W` appear in `bncCite`. However, where records in subordinate tables (`bncAnalytic`, `bncAuthEd`, `bncIdno`, `bncPermission`, `bncSample`, `bncSerial`, `bncTextNote`) can join to records either in `bncText` or in `bncCite`), the *xx*`CiKey` field has a value of `W` if the join is to `bncText`, `B` or `S` if to `bncCite`.

Since additional citations may occur only for spoken texts, and the database contains no information about spoken texts as yet, it contains no information about additional citations.

```
create table bncCite
(
    ciTeKey         char(6)           not null, /* Unique text key      */
    ciKey           char(1)           not null, /* See above            */

    /* Note that, in contrast to teTitle above, ciTitle may be null:    */
    /* no generated title should be given in the event that the         */
    /* monographic work is untitled.                                    */

    ciTitle         varchar(128),               /* Monographic title    */
    ciImKey         char(6),                    /* imprint (if any) for */
                                                /*   monographic text   */
    /* See notes for teImprintPerm                                      */

    ciImprintPerm   char(1),                    /* Perms from imprint?  */

    ciSeKey         char(6),                    /* Serial (if any) of   */
                                                /*   which this is part */
    /* See notes for teDate                                             */

    ciDate          varchar(21)                 /* Publication/         */
                                                /*   origination date   */
);
create index ciIndex1 on bncCite(ciTeKey);
create unique index ciIndex2 on bncCite(ciTeKey, ciKey);
```

3.4.4 bncEditText

This table contains one record for each different editorial practice applying to an individual text in the corpus (except for practices applying to a whole class of texts — see bncEditorial). The information is used in the header editorial practices declaration.

The database currently contains no information about editorial practices.

```
create table bncEditText
(
    etTeKey         char(6)           not null, /* Keys a text          */
    etPrKey         char(1)           not null, /* Type of practice     */
    etEdKey         integer           not null  /* Keys a practice      */
);
create index etIndex1 on bncEditText(etTeKey);
create unique index etIndex2 on bncEditText(etTeKey, etPrKey, etEdKey);
```

3.4.5 bncIdNo

This table records identifying numbers for analytic, monographic or serial texts, with the exception of ISBN (monographic texts) and ISSN (serials), which are held in `teISBN` and `seISSN` respectively; and the local identifier, held in `teKey`.

There is currently no information about identifying numbers in the database.

```
create table bncIdNo
(
    idTeKey         char(6)           not null, /* Keys a text           */
    idCiKey         char(1)           not null, /* Type of citation      */

    /* Following field is zero where information relates to a            */
    /* monographic text; negative for a serial; positive for an analytic*/
    /* text.                                                             */

    idAnKey         smallint          not null, /* Analytic text no.     */

    idItKey         char(6)           not null, /* Type of idno          */
    idValue         varchar(16)       not null  /* Value of idno         */
);
create index idIndex1 on bncIdNo(idTeKey);
create index idIndex2 on bncIdNo(idTeKey, idCiKey, idAnKey);
```

3.4.6 bncImGrant

This table contains one record for each distinct imprint (publisher, distributor, release authority) or permissions grantor pertaining to works in the corpus. Where permissions are granted by an author or editor, the name of that person appears both here and in bncAuthEd.

Note that this information is normalized: it is necessary to do this, as a single permissions grantor may grant permissions covering many corpus texts. Depending on the contents of teImKey, ciImKey and the presence or absence of bncPermissions records, the information appears either in a structured citation (source, script or broadcast statement) in a text header; in the permissions information in the source statement; or in both places.

There is currently no information about imprints or grantors in the database, other than that in the bncOUPdb table. This will be transferred semi-automatically to the bncAuthEd table in the near future.

```
create table bncImGrant
(
    imKey           char(6)           not null, /* Unique key            */
    imName          varchar(64)       not null, /* Grantor name          */
    imPlace         varchar(32),                /* City, town, village   */
    imTerritory     varchar(32),                /* Territory covered     */
    imEnds          date,                       /* When grant ends       */
    imNotice        varchar(512)                /* Notice relating to    */
                                                /*   grant               */
);
create unique index imIndex1 on bncImGrant(imKey);
```

3.4.7 bncLangUsage

This table contains one record for each language used in each corpus text. It is not currently in use. Its format is subject to change because I'm not certain about the shape of laKey.

The information is used in creating the language usage description in text headers.

```
create table bncLangUsage
```

```
(
    luTeKey     char(6)               not null, /* Keys a text          */
    luLaKey     char(2)               not null, /* Keys a language      */
    luUsKey     char(1)                         /* Keys usage amount    */
);
create index luIndex1 on bncLangUsage(luTeKey);
```

3.4.8 bncOUPdb

This table containing fields corresponding to those appearing in d BASE reports accompanying texts captured by OUP. The names and presence of certain fields are intuited as this information does not appear in materials received to date. Note that `ouKey` should join `bncText` for a single publication, `bncSerial` for a serial. It will not join either if the record describes a number of related texts containing ephemeral items.

The information is not directly used in the creation of headers.

The table is currently populated with information about all texts received from OUP to date, except those derived from machine-readable newspaper material. It should be possible to transform and transfer the information in this table so as to fill in fields in other database tables.

```
create table bncOUPdb
(
    ouKey           char(6)           not null, /* Six-letter T code    */
    ouAcquisit      varchar(32),                /* Acquisition mode,    */
                                                /*   date               */
    ouAuthAge       char(16),                   /* Author age           */
    ouAuthDom       char(8),                    /* Author domicile      */
    ouAuthEth       char(16),                   /* Author ethnic origin */
    ouAuthMode      char(16),                   /* Author mode (single, */
                                                /*   multiple etc.)     */
    ouAuthSex       char(8),                    /* Author sex           */
    ouBibAuth       varchar(64),                /* Author(s), editor(s) */
    ouBibDate       varchar(32),                /* Publication date(s)  */
    ouBibOrigDate   char(8),                    /* Orig. pub. date      */
    ouBibPlace      varchar(32),                /* Place of publication */
    ouBibPubl       varchar(64),                /* Publisher            */
    ouBibTitl       varchar(64),                /* Title                */
    ouCirculatn     char(16),                   /* Circulation          */
    ouFormComp      char(16),                   /* Single/comp'te/coll'n*/
    ouFormSampext   char(16),                   /* Sample end points    */
    ouFormSampSize  char(16),                   /* Words in sample      */
    ouFormTextsize  char(16),                   /* Pages in original    */
    ouGenre         varchar(32),                /* Genre (6 choices)    */
    ouId            integer,                    /* Identifying number   */
    ouInform        char(16),                   /* {Inform,Imagina}tive */
    ouLevel         char(8),                    /* Audience level       */
    ouOrigpub       varchar(64),                /* Original publisher   */
    ouSubjDomain    varchar(32),                /* Domain (9 choices)   */
    ouSubjFields    varchar(64),                /* Subject keywords     */
    ouTargetAge     char(8),                    /* Target age (3 choices*/
    ouTargetSex     char(8)                     /* Target sex           */
);
create index ouIndex on bncOUPdb(ouKey);
```

3.4.9 `bncPermission`

This table contains one record for each link between an analytic, monographic, or serial text and a permissions grant; that is, it joins records in `bncImGrant` either to `bncText`, `bncAnalytic`, or `bncSerial`, or to `bncCite`. It is used in generating permissions information for the header source description where permissions are not solely controlled by the person or organization named as the imprint.

The database currently holds no permissions information.

```
create table bncPermission
(
    peTeKey         char(6)           not null, /* Keys a text           */
    peCiKey         char(1)           not null, /* Type of citation      */

    /* Following field is zero where information relates to a            */
    /* monographic text; negative for a serial; positive for an analytic*/
    /* text.                                                             */

    peAnKey         smallint          not null, /* Analytic text no.     */

    peImKey         char(6)           not null  /* Keys grantor          */
);
create index peIndex1 on bncPermission(peTeKey);
create index peIndex2 on bncPermission(peTeKey, peCiKey, peAnKey);
create unique index peIndex3 on bncPermission(peTeKey, peCiKey,
                                              peAnKey, peImKey);
create index peIndex4 on bncPermission(peImKey);
```

3.4.10 `bncRecording`

This table contains at least one record for each corpus text with which an audio recording is associated. There is more than one record if more than one named person or organization is associated with the recording. This involves redundant duplication of `reRmKey` if multiple records are necessary; however, the expectation is that this will seldom be the case.

The information is used in the header recording statement.

The database currently contains no information about recordings.

```
create table bncRecording
(
    reTeKey         char(6)           not null, /* Key to text           */
    reRmKey         char(6)           not null, /* Recording method key */
    reWho           varchar(64),                /* Associated person or */
                                                /*   institution         */
    reRole          varchar(64)                 /* What they did         */
);
create index reIndex1 on bncRecording(reTeKey);
```

3.4.11 `bncRegClas`

This table contains one record for each open regional classification which may be given for each corpus text. Classifications for which no value has yet been given have a null `seRgKey`. The types of regional classification applicable to a particular class of text are set out in `bncFeatSpec`. Examples are author domicile, and speaker birthplace, both used in text classification.

All texts for which information has been loaded into the database have corresponding `bncRegClas` records, but all of these currently have null `rcRgKey`s.

```
create table bncRegClas
(
    rcTeKey         char(6)           not null, /* Keys bncText          */
    rcFeClass       char(6)           not null, /* Keys bncFeatures      */
    rcRgKey         char(3)                     /* Keys bncRegions       */
);
create index rcIndex1 on bncRegClas(rcTeKey);
```

3.4.12 `bncRevDesc`

Contains one or more records for each change note associated with an entry in bncStatus. For long notes, the `reNote` fields from successive records, ordered by `rdKey`, are concatenated.

This information, used in generating header revision descriptions, may be generated in an as-yet-undefined manner from the `Z_` information files associated with each text. The table is not currently used.

```
create table bncRevDesc
(
    rdTeKey         char(6)           not null, /* Keys a text           */
    rdSdKey         char(1)           not null, /* Keys a status         */
    rdKey           integer1          not null, /* Makes key unique      */
                                                /*   (counts up from 1) */

    /* Following fields may be null if note info is the same as status  */
    /* change info, or if rdKey <> 1                                    */

    rdDate          date,                       /* Date of note          */
    rdWho           char(8),                    /* Noter (login name or */
                                                /* institution initials)*/

    rdNote          varchar(256)      not null  /* Note text             */
);
create index rdIndex1 on bncRevDesc(rdTeKey, rdSdKey);
create unique index rdIndex2 on bncRevDesc(rdTeKey, rdSdKey, rdKey);
```

3.4.13 `bncSample`

This table contains records for each text which is a sample. Page numbers are entered as printed, less ornamentation: `xix`, `213`, `5.27`, and so on. Since the size of a sample may change as a result of processing (for example, when a long sample is cut to 40,000 words), the record having the highest value of `saSdKey` is the most current. `saNote` will typically be empty (null). Nulls are allowed elsewhere so that information can be filled in after a record has been created.

This information is used in the header sampling declaration. It is currently present in the database for sampled texts which have been processed to the `B_` level, and which contain the relevant information in a standard format in their dummy headers. Sampling information also exists in `bncOUPdb` records; it should be possible to use this to create `bncSample` records for dot (original) text files.

```
create table bncSample
(
    saTeKey          char(6)            not null, /* Unique text key      */
    saSdKey          char(1)            not null, /* Status to which this */
                                                  /*   information applies*/
    saCiKey          char(1)            not null, /* Citation type (likely*/
                                                  /*   always to be W)    */
    saBegSrc         char(8),                     /* Source beginning page*/
    saEndSrc         char(8),                     /* Source ending page   */
    saBegSamp        char(8),                     /* Sample beg. page     */
    saEndSamp        char(8),                     /* Sample end page      */
    saNote           varchar(256)                 /* Notes on sample      */
);
create index saIndex1 on bncSample(saTeKey);
create unique index saIndex2 on bncSample(saTeKey, saCiKey, saSdKey);
```

3.4.14 `bncSelClas`

Contains one record for each closed-class classification or selection feature which must be given (selection features) or may be given (classification features) for each corpus text. Note that the values of some features (for example, sample type) may change as a text proceeds through the "sausage machine". However, no historical record of feature values is kept: only information about the current version of the text (or rather, the text as it was when the feature values were last set) is kept. All records required for a particular text will typically be added with null values for `scFeCode` when the text is introduced to the database. These nulls are replaced with values as these are established. The records required for a text of a given class are established by look-up in `bncFeatSpec`.

The information is used in the header's text classification. It is intended that `scFeClass` and `scFeCode` vales can be concatenated to give values for `target` IDREFS for `<catRef>` tags.

Information is currently present (and non-null) in the database in two cases:

- Texts derived from machine-readable newspapers are classified as to their domain; and
- Texts which have reached the `B_` stage, and which have the information in a standardized format in their dummy headers, are classified as beginning, middle or end samples etc.

Classification information exists in the `bncOUPdb` table, and should be usable in filling in currently-null `scFeCode` values.

```
create table bncSelClas
(
    scTeKey          char(6)            not null, /* Keys bncText          */
    scFeClass        char(6)            not null, /* Keys bncFeatures      */
    scFeCode         integer1                     /* Keys bncFeatures      */
);
create index scIndex1 on bncSelClas(scTeKey);
create unique index scIndex2 on bncSelClas(scTeKey, scFeClass);
```

3.4.15 bncSerial

This table contains one record for each serial publication represented in the BNC. Note that a serial should always have an imprint; however, nulls are allowed so that this information can be added after a record has been created.

Information about serials will generally appear in the header source description. although it may also pop up in script or broadcast statements.

No information about serial publications is yet recorded in the database, even though many of the monographic texts in the corpus are derived from issues of serial publications.

Note that seKey occupies the same name-space as teKey: a text and a serial may not have the same six-character key. For example, the serial BestMg corresponds to the BNC texts BestMA, BestMB, and so on — there can be no text called BestMg. (Or, indeed, given the mono-case restriction on uniqueness, BestMG.)

```
create table bncSerial
(
    seKey           char(6)           not null, /* Six character code   */
    seTitle         varchar(128)      not null, /* Periodical title     */
    seISSN          char(8),                    /* ISSN (if any).  No   */
                                                /*   spaces, minuses.   */
    seImKey         char(6),                    /* Keys an imprint      */

    /* The following field has value ``Y'' if world-wide permissions    */
    /* controlled by agency named as imprint; ``N'' if permissions are  */
    /* known to be controlled by one or more agencies in addition to or */
    /* instead of the imprint agency; ``U'' if permissions status is    */
    /* unknown.  If the value is ``N'', there should be records in      */
    /* bncPermission joining this record.   Field is null if seImKey is */
    /* null.                                                            */

    seImprintPerm   char(1)                     /* Perms from imprint?  */
);
create unique index seIndex1 on bncSerial(seKey);
```

3.4.16 bncStatus

This table contains one record for each status change for each text. That is, there is a record for the dot (original) file; a record for the A_ file; a record for the B_; and so on. The meaning of each status, and any necessary follow-up, is briefly described in the bncStatusDesc table.

Most status information is not used in the text header; however, word count and file size information from the final (E_) version of a file appear in the extent statement, and version and revision information in the edition statement. Conceivably, status information could be used in generating revision descriptions. (See also bncrevDesc.)

Status information currently exists in the database for all dot (original) files received from OUP, and for A_ and B_ files present on the system at 17:00 on 24th September.

```
create table bncStatus
(
    stTeKey         char(6)           not null, /* Unique text key      */
    stSdKey         char(1)           not null, /* Status code (see     */
                                                /*   comment in bncText)*/
    stDate          date              not null, /* Status change date   */
    stWho           char(8)           not null, /* Changer (login name) */

    /* Where a status change requires a follow-up, such as archiving or */
    /* accession to the corpus, this field is set to the date on which  */
    /* the follow-up occurred.                                          */

    stFollowUp      date,                       /* Date followed up     */

    /* The following field records the number of words in a text at the */
    /* time of a status change.  The length of a text may vary as       */
    /* processing proceeds because of deletions, additions, corrections */
    /* etc.                                                             */

    stWords         integer,                    /* Words in text        */

    stK             integer,                    /* Kilobytes for text   */

    /* The following fields are null until a text has been accessioned  */
    /* into the BNC.  At that time, stTextRevision becomes 1, and       */
    /* stCorpVersion and stCorpRevision are set to indicate the release */
    /* level of the corpus in which the text first appears.  Subsequent */
    /* revisions to a text are reflected in changes to these fields.    */

    stTextRevision  integer1,                   /* Text revision no.    */
    stCorpRevision  integer1,                   /* Appears in this rev. */
    stCorpVersion   integer1                    /* of this BNC version  */
);
create index stIndex1 on bncStatus(stTeKey);
create unique index stIndex2 on bncStatus(stTeKey, stSdKey);
```

3.4.17 bncTags

This table contains one record for each type of tag (excluding header tags) used in a particular corpus text. taWho and taNote are expected normally to be null. Note that information about a given tag appears only once for each text; there is no provision for keeping track of tag usage at particular stages in a text's processing. It is expected that bncTags records for a given text will be generated at the time of accession to the corpus. (E_ stage.)

The information is used in the header's tag usage section. It does not currently exist for any text in the database.

```
create table bncTags
(
    taTeKey     char(6)               not null, /* Keys a text          */
    taTdKey     char(8)               not null, /* Keys a tag           */
    taCount     integer               not null, /* Count for this tag   */
    taWho       char(8),                        /* Person writing note  */
                                                /*   (logname)          */
    taNote      varchar(256)                    /* Note text            */
);
create index taIndex1 on bncTags(taTeKey);
create unique index taIndex2 on bncTags(taTeKey, taTdKey);
```

3.4.18 bncTextNote

This table contains notes about an analytic, monographic or serial text. Where a note is too long to fit in a single record, `tnNote` fields are concatenated in the sequence dictated by `tnSeq`, which starts from 1.

The information appears as notes in structured citations in the text header. There is currently no information of this type in the database.

```
create table bncTextNote
(
    tnTeKey         char(6)           not null, /* Keys a text           */
    tnCiKey         char(1)           not null, /* Type of citation      */

    /* Following field is zero where information relates to a           */
    /* monographic text; negative for a serial; positive for an analytic*/
    /* text.                                                             */

    tnAnKey         smallint          not null, /* Analytic text no.     */

    tnSeq           smallint          not null, /* Sequence number       */
    tnNote          varchar(256)      not null  /* Note text             */
);
create index tnIndex1 on bncTextNote(tnTeKey, tnCiKey);
create index tnIndex2 on bncTextNote(tnTeKey, tnCiKey, tnAnKey);
```

3.4.19 bncUnique

This table contains one record for each monographic text or serial in the corpus. `unKey` is an upper-case version of `unTeKey`, and is used to enforce uniqueness of keys. (There are many ways to skin this cat. This one will do.)

Unique key information exists in the database for all monographic texts received from OUP. No information yet exists for serials.

```
create table bncUnique
(
    unKey       char(6)               not null, /* Monocase key          */
    unTeKey     char(6)               not null  /* Mixed-case key        */
);
create unique index unIndex1 on bncUnique(unKey);
```

3.5 Look-up tables

Unless otherwise stated, all of the tables listed in this section are currently populated in the database.

3.5.1 bncEditorial

This table contains one record for each distinct editorial practice used in the corpus; that is, there is one record for each type of normalization practice; one for each type of hyphenation practice, and so on. The information is used in building the header editorial practices declaration. It is not currently available in the database.

```
create table bncEditorial
(
    edPrKey         char(1)           not null, /* Class of practice    */
    edKey           integer           not null, /* Makes key unique     */

    /* Where an editorial practice applies to all texts of a particular */
    /* class (for example, spoken demographic), the  following field    */
    /* takes the values enumerated for fsTeType.  Where the practice    */
    /* does not apply to a whole class, the field is null.              */

    edTeType        char(3),                    /* Applies to text type */
    edDesc          varchar(1024)     not null  /* Description          */
);
create unique index edIndex1 on bncEditorial(edPrKey, edKey);
```

3.5.2 bncFeatSpec

This table specifies the selection and classification features applicable to a particular text type. It is used in creating `bncRegClas` and `bncSelClas` records for a text.

```
create table bncFeatSpec
(
    fsTeType        char(3)           not null, /* Applicable to type...*/
    fsClass         char(6)           not null, /* Class code           */
    fsSelClass      char(1)           not null, /* Selection/Classific- */
                                                /*   -ation (S/C)       */
    fsFeatReg       char(1)           not null  /* Closed class (F) or  */
                                                /*   regional (R) type? */
);
```

3.5.3 bncFeatures

This is a look-up table for closed sets of selection and classification features. Note that, if `feCode` is zero, `feName` gives class name. It is used in expanding small numeric codes into human-readable forms.

```
create table bncFeatures
(
    feClass         char(6)          not null,  /* Short class name     */
    feCode          integer1         not null,  /* Sub-class code       */
    feName          varchar(64)      not null   /* Sub-class name       */
);
create unique index feIndex1 on bncFeatures(feClass, feCode);
```

3.5.4 bncIdType

This table lists types of idno used in the corpus. It is used in expanding short codes into descriptions.

```
create table bncIdType
(
    itKey           char(6)           not null, /* Code type            */
    itName          varchar(64)       not null  /* Code name            */
);
create unique index itIndex1 on bncIdType(itKey);
```

3.5.5 bncLanguage

This table contains a record for each language or language variant recognized by the corpus. (Layout is provisional, pending further findings on language naming standards. For the moment it accommodates only ISO 639:1988.)

```
create table bncLanguage
(
    laKey       char(2)               not null, /* Unique key           */
    laName      varchar(32)           not null  /* Language name        */
);
create unique index laIndex1 on bncLanguage(laKey);
```

3.5.6 bncPractices

This table contains one record for each class of editorial practice used in the corpus. (Normalization, hyphenation, etc.)

```
create table bncPractices
(
    prKey           char(1)           not null, /* Unique key           */
    prBNCName       char(8)           not null, /* BNC tagname          */
    prFullName      varchar(16)                 /* Full name            */
);
```

3.5.7 bncRecMethod

This table contains one record for each distinct recording method used in the capture of spoken texts. (Walkman operated by demographic recruit; DAT recorder operated by Longman staff etc.) The division of data between this table and bncRecording assumes that there will be far fewer distinct recording methods than actual recordings.

The table is not currently populated.

```
create table bncRecMethod
(
    rmKey           char(6)           not null, /* Unique key           */
    rmType          char(1)           not null, /* Audio/video (A/V)    */
    rmEquipment     varchar(128)      not null  /* Description of method*/
);
```

3.5.8 bncRegions

This is a look-up table for regions used in classification features. Its contents are based on country codes from ISO 3166:1988 and revisions. Private-use codes are used for BNC-specific region names.

```
create table bncRegions
(
    rgCode          char(3)           not null, /* Region code          */
    rgName          varchar(64)                 /* Region name          */
);
create unique index rgIndex1 on bncRegions(rgCode);
```

3.5.9 `bncStatusDesc`

This table contains one record for each text status used in the BNC text processing procedures. (`.`, `A_` etc.)

```
create table bncStatusDesc
(
    sdKey           char(1)          not null, /* Status code          */
    sdDesc          varchar(64)      not null, /* Description of status*/
    sdFollowUp      varchar(64)      not null  /* Desc. of follow-up   */
);
create unique index sdIndex1 on bncStatusDesc(sdKey);
```

3.5.10 `bncTagDesc`

This table contains one record for each type of element allowed in corpus texts, excluding header elements. The current database reflects tags allowed by CDIF 1.1.

```
create table bncTagDesc
(
    tdKey       char(8)              not null, /* Tagname              */
    tdEmpty     char(1)              not null, /* Empty (Y/N)          */
    tdType      char(1)              not null  /* Required (Mandatory)/*/
                                               /*   Recommended/       */
                                               /*   Optional (M/R/E)   */
);
create unique index tdIndex1 on bncTagDesc(tdKey);
```

3.5.11 `bncUsage`

This table contains one record for each language usage value allowed in corpus texts. (Low, 10%, 20%...)

```
create table bncUsage
(
    usKey       char(1)              not null, /* Unique key           */
    usDesc      varchar(16)          not null  /* Expanded form        */
);
create unique index usIndex1 on bncUsage(usKey);
```

4 Example Queries

Members of the `xzug` group with accounts on the BNC Suns can query the database. For the moment, only I can amend its contents: I want to put some integrity constraints on it, and give it a proper user interface before I throw it open to all-comers. Please bear with me.

To get into the database, use one of

`sql bnc` Nasty, line-orientated interface to SQL.

`ingmenu bnc` Clunky, forms- and screen-based interface. Gives a choice of database utilities, including SQL.

`isql bnc` Clunky, screen-based interface to SQL only.

The following subsections present some queries that work. Before trying them, tell *Ingres* to `set autocommit on`, otherwise your whole session is treated as a single transaction, locking other users out of areas of the database and eventually overflowing the transaction log staging area.

4.1 Find out how much text has been delivered to OUCS:

```
* select sum(terecwords) from bnctext
* \go
Executing . . .

+-------------+
|col1         |
+-------------+
|     23324317|
+-------------+
(1 row)
```

or

```
* select sum(stwords) from bncstatus where stsdkey = '.'
* \go
Executing . . .

+-------------+
|col1         |
+-------------+
|     23324317|
+-------------+
(1 row)
```

The *Ingres* terminal mointor, `sql` was used to enter these queries.

4.2 Find out how much text OUP delivered on 29th June:

```
* select sum(terecwords) from bnctext where
* terecfrom = 'OUP' and terecdate = '920629'
* \go
Executing . . .

+-------------+
|col1         |
+-------------+
|      6592937|
+-------------+
(1 row)
```

4.3 Find out how much text has been processed to the `B_` stage:

```
* select sum(stwords), max(stdate) from bncstatus
* where stsdkey = 'B';
* \go
Executing . . .

+-------------+------------------------+
|col1         |col2                     |
+-------------+-------------------------+
|      8385018|920924                   |
+-------------+-------------------------+
(1 row)
```

(Note the ISO-format date.)

4.4 Find out about the text `SexDis`:

```
* select tekey, tesdkey, tetitle from bnctext
* where tekey = 'SexDis';
* \g
Executing . . .

+------+------+------------------------------------
|tekey |tesdke|tetitle
+------+------+------------------------------------
|SexDis|B     |SEXUAL DISSIDENCE JONATHAN DOLLIMORE
+------+------+------------------------------------
(1 row)
```

(This wraps on the screen, as titles are allowed to be very long. It also shows that titles pulled out of texts need cleaning up by hand — this one erroneously includes the author's name.)

4.5 Find the distinct features known for an edition of *The Guardian*:

```
* select distinct scfeclass, scfecode, fename
* from bncselclas, bncfeatures
* where sctekey like 'Ga%' and scfecode is not null
* and scfeclass = feclass and scfecode = fecode
* \g
Executing . . .

+------+------+----------------------------------
|scfecl|scfeco|fename
+------+------+----------------------------------
|wriDom|     4|Informative -- social science
|wriDom|     5|Informative -- world affairs
|wriDom|     6|Informative -- commerce & finance
|wriDom|     7|Informative -- arts
|wriDom|     9|Informative -- leisure
+------+------+----------------------------------
(5 rows)
```

4.6 Find the relative percentages of each domain:

```
  2> create table tmpwords as
  3> select sum(stwords) as sumstwords
  4> from bncstatus, bncselclas
  5> where sttekey = sctekey
  6> and stsdkey = 'B'
  7> and scfeclass = 'wriDom'
  8> and scfecode is not null

(1 row)

  2> create table tmpdomain as
  3> select sum(stwords) as sumdomain, fename
  4> from bncstatus, bncselclas, bncfeatures
  5> where sttekey = sctekey
  6> and stsdkey = 'B'
  7> and scfeclass = 'wriDom'
  8> and scfeclass = feclass
  9> and scfecode = fecode
 10> group by fename

(8 rows)

  2> select 100.00*sumdomain/sumstwords as percentage, fename
  3> from tmpdomain, tmpwords

+-----------+-------------------------------------
|percentage |fename
+-----------+-------------------------------------
|      0.570|Informative -- applied science
|     11.538|Informative -- arts
|      0.180|Informative -- belief & thought
|     14.432|Informative -- commerce & finance
|     25.424|Informative -- leisure
|      2.575|Informative -- natural & pure science
|      2.286|Informative -- social science
|     42.995|Informative -- world affairs
+-----------+-------------------------------------
(8 rows)

  1> drop table tmpwords

  2> drop table tmpdomain

End of Request
```

Various qualifiers are used to select only those texts for which the information is known. The information in the database currently relates only to newspaper material. There may be a more direct means of making this query, but my SQL is not up to it.

The *Ingres* screen-based SQL interface, `isql`, was used to enter this query.

4.7 Find relative percentages for each type of sample:

```
  2> create table tmpwords as
  3> select sum(stwords) as sumstwords
  4> from bncstatus, bncselclas, bncfeatures
  5> where sttekey = sctekey
  6> and stsdkey = 'B'
  7> and scfeclass = 'wriSam'
  8> and scfecode is not null
  9> and fename like '%sample%'
 10> and scfeclass = feclass
 11> and scfecode = fecode

(1 row)

  2> create table tmpsample as
  3> select sum(stwords) as sumdomain, fename
  4> from bncstatus, bncselclas, bncfeatures
  5> where sttekey = sctekey
  6> and stsdkey = 'B'
  7> and scfeclass = 'wriSam'
  8> and fename like '%sample%'
  9> and scfeclass = feclass
 10> and scfecode = fecode
 11> group by fename

(3 rows)

  2> select 100.00*sumdomain/sumstwords as percentage, fename
  3> from tmpsample, tmpwords

+-----------+--------------------------------
|percentage |fename
+-----------+--------------------------------
|     36.289|Beginning sample
|     32.826|End sample
|     30.885|Middle sample
+-----------+--------------------------------
(3 rows)
End of Request
```

Again, only texts for which the information is known are selected.