NATIONS UNIES
COMMISSION ECONOMIQUE POUR L'EUROPE

ОБЪЕДИНЕННЫЕ НАЦИИ
ЭКОНОМИЧЕСКАЯ КОМИССИЯ ДЛЯ ЕВРОПЫ

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

SEMINAIRE | СЕМИНАР | SEMINAR

STATISTICAL COMMISSION AND
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN
STATISTICIANS

Distr.
GENERAL

CES/SEM.38/7
24 April 1998

ENGLISH ONLY

Seminar on integrated statistical information systems and related matters (ISIS '98)
(Geneva, Switzerland, 27-29 May 1998)

Topic (i): use of communication technologies for external and internal disseminationn of statistical data

SEARCH ENGINE FOR STATISTICAL DATABASES

Submitted by the United Nations Statistical Division[1]

1. Using the relational model, statistical databases are often designed with a single, multidimensional fact table and several reference tables, one for each dimension of the data or fact table. Typical examples of reference or dimension tables are periodicity, location and series. The periodicity table may contain the codes of the periods together with some textual description of their meaning and possibly other attributes such as the duration of the period. The location table typically consists of location code, location name and possibly some additional attributes describing in more details the coverage of the corresponding location. The series table again consists of a series code and one or more text attributes describing the meaning of the code. In the United Nations Statistics Division databases and the United Nations Economic and Social Information System (UNESIS), the periodicity tables typically contain years and the location tables describe countries. The series table may have entries such as "Population by type of residence (urban/rural), sex and age" and "Gross fixed capital formation (SNA68) in national currency, current prices".
Using the entity/relationship methodology, such databases are usually modeled with "star" schemas. The database model consists of one fact table, often quite large, surrounded by much smaller dimension tables, each one contributing key attributes to the fact table (see figure 1).

2. The Series table, which tends to be larger in both number of rows and in width of the description column, is used in the examples below. The examples are coded in Sybase Transact-SQL with JYACC's JPL (JYACC Procedural Language) for the client-side programming. JPL is a programming dialect similar to C.

---

[1] Prepared by Lubomir Vitkov.

3. The idea is to programmatically prepare an "index" table, containing a row for each word contained in the description

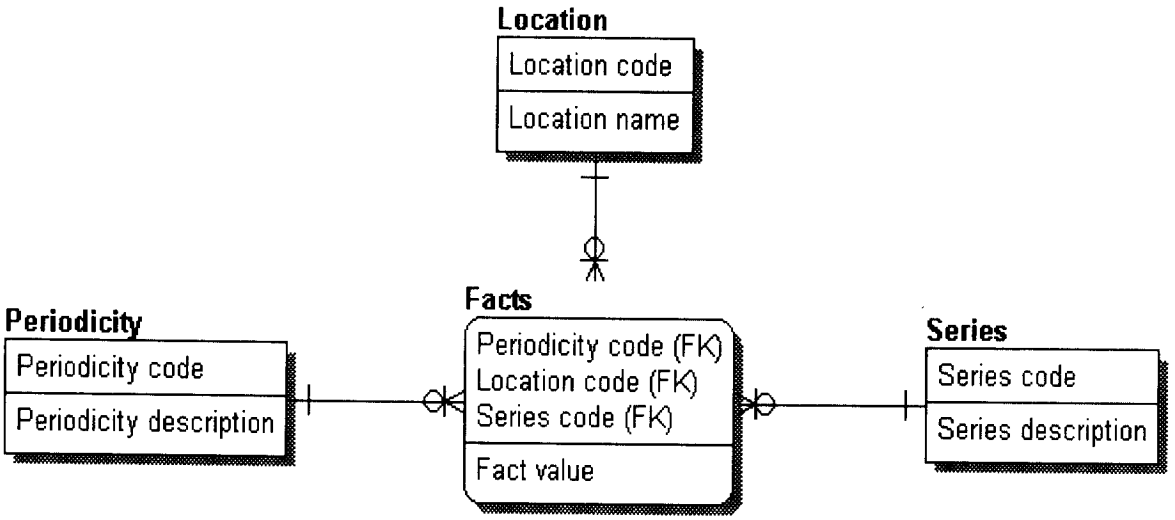

Figure 1

columns. The index table will also contain the series code of the series where the word occur and the soundex[2] code of the word. To ensure a unique primary key of the table, probably the simplest way is to introduce an identity column, which Sybase keeps unique automatically:

```
CREATE TABLE word_index
(
      word                char(20),
      series_code int,
      soundex_code        char(4)
      seq_no        = identity(5)
)
go
ALTER TABLE word_index
  ADD CONSTRAINT pk_word_index PRIMARY KEY (seq_no)
Go
```

4. The word index will obviously be searched by soundex code, so the performance will be greatly improved by an index on this column:

```
CREATE UNIQUE NONCLUSTERED INDEX index_on_soundex
   on word_index (soundex_code)
go
```

5. Having defined the necessary database structure, let's turn to some client side programming and populate the word index table. What we need is a program which takes the descriptions one by one, breaks them into words and stores each word in the word

---

[2] Soundex is an algorithm which assigns identical, four digits codes to "similarly" sounding words. More information can be found at **Error! Reference source not found.** and other Web sites. Numerous implementations are also available on the Internet.

index table, together with the code of the corresponding series and the soundex code of the word. The programme first reads the code and the text description columns in memory arrays:

```
SELECT series_code as code, LOWER(series_description) as description FROM series
```

6. If the source table is very large and cannot be read entirely in the program running on the client machine, then cursor techniques to read smaller number of rows have to be employed. The different client authoring systems and the different DBMS's support variety of such techniques.

7. The next step is to organize two nested loops -- the outer one cycling through the instances of the description, and the inner one breaking the description string into separate words. The description string is scanned character by character and words are defined as substrings delimited by space before and after the substring. In the process any "noise" characters such as extra spaces, punctuation characters, dollar sign, percent sign etc. are disregarded. Whenever a new word is produced by the inner loop, it is stored in the word index table:

```
INSERT INTO word_index (word,series_code,soundex_code)
VALUES (
      ":current_word",
      :current_series,
      SOUNDEX(":current_word")
)
```

where current_word and current_series are de-referenced program variables.
The above INSERT uses the Sybase SOUNDEX function, which yields quite satisfactory results for texts in English. For different languages, especially languages using different alphabets, custom soundex functions may have to be developed, executed as part of the client program and the result passed to the DBMS as four-character string. The frequency distribution of some sample data is given in Appendix 1.

8. Next we will describe how the word index table is used for the actual search. We present the user with a very simple screen, designed along the lines of the Internet search sites:

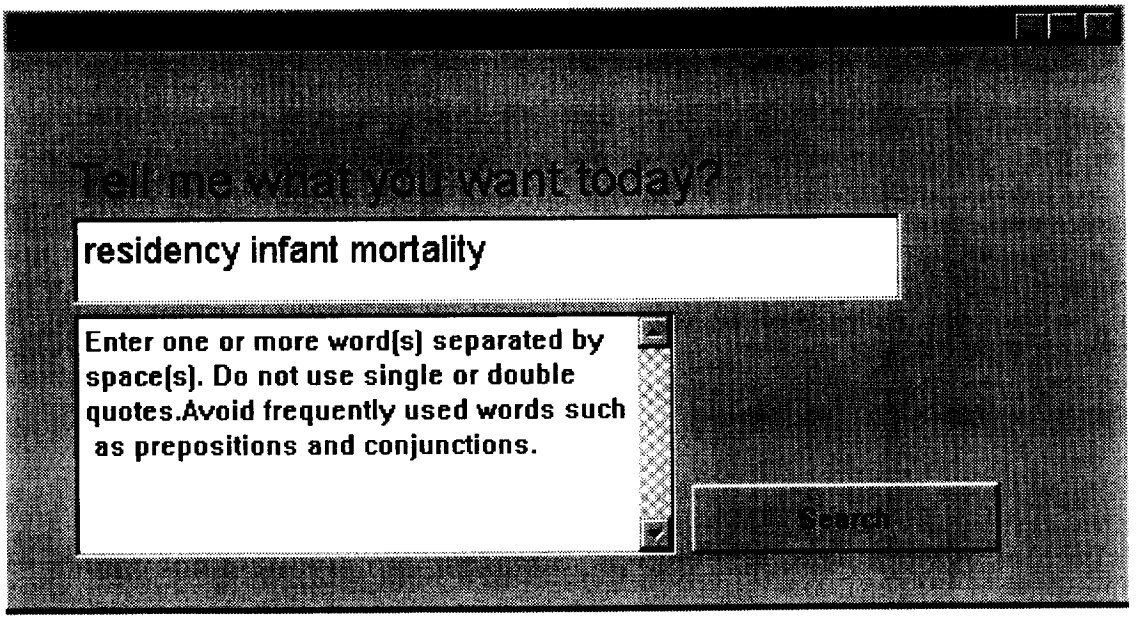

Figure 2

9. Using the technique described above, the programme will break the search string into individual words and prepare a new string looking like this:

```
(soundex("residency"), soundex("infant"), soundex("mortality")) (1)
```

This new string can be used as a parameter in an SQL SELECT statement. For the "series" table from figure 1, then we can write

```
Select distinct series.* from series s, word_index i
      where s.series_code=i.series_code and          (2)
            i.soundex_code in :search_string
```

The search_string variable contains of course the string (1) above. The SELECT (2) will return all the series whose description contains at least one of the search words. The DISTINCT modifier prevents the same series row to be selected more than once in the cases where the series description contains more than one of the search words.

10. It is perfectly possible to search for series whose descriptions contain ALL of the search words. Some new controls have be added to the screen in figure 2, so that the user can indicate "AND" or "OR" mode of the search. Yahoo[3] for example, uses radio buttons labeled "Matches on all words (AND)" and "Matches on any word (OR)" for this purpose. Alternatively, the user may be asked to combine the search words with plus and minus signs and search for texts containing the words prefixed by plus and not containing the words prefixed by minus. The parsing and the execution of such expressions may become more complex and is outside of the scope of the present paper. In the simpler case of an entirely "AND" search the SQL we pass to the database engine will look like:

```
select series.*
      from  series
      where series_code in (
      select a.series_code
      from word_index a, word_index b, word_index c
      where a.series_code  = b.series_code          (3)
        and a.series_code  = c.series_code
        and a.soundex_code = soundex("residency")
        and b.soundex_code = soundex("mortality")
        and c.soundex_code = soundex("infant"))
```

11. To prepare the IN list we are using a non-correlated sub-query where the word index table is self-joined as many times as search words are submitted. Different DBMS systems have different restrictions on the number of tables that can be joined together, which means that the number of search words that can be submitted to an AND-type search is limited. In Sybase SQL server this restriction is 16. Even such a heavy duty RDBMS such as NCR's Teradata has a limitation of a maximum of 64 tables in a single join[4]. To the best knowledge of the writer, Oracle is the only RDBMS that does not have a restriction on the number of tables participating in a join. It should be noted however that, as the number of tables in the join increases, the performance of the DBMS, generally speaking, decreases.

---

[3] **Error! Reference source not found.**

[4] Walter, T. "What's in store", *Teradata review*, 1(1), Spring 1998.

12. The following alternative formulations of the SELECT (3) above were kindly proposed by some of the first readers[5] of this paper:

```
select distinct  s.series_description
      from  series s,
            word_index a,                    (4)
            word_index b
            word_index c
where s.series_code = a.series_code              and
      s.series_code = b.series_code              and
      s.series_code = c.series_code              and

      a.soundex_code  = soundex("residency")     and
      b.soundex_code  = soundex("mortality")     and
      c.soundex_code  = soundex("infant ")
```

And

```
select distinct  s.series_description
from series s
where s.series_code in ( select  t1. series_code
                         from word_index t1 where
                         t1.soundex_code = soundex("residency") )
and   s.series_code in ( select  t1. series_code                    (5)
                         from word_index t1 where
                         t1.soundex_code = soundex("mortality") )
and   s.series_code in ( select  t1. series_code
                         from word_index t1 where
                         t1.soundex_code = soundex("infant") )
```

13. The SELECT (4) does not use correlated sub-query and we may expect it to perform better than (4) since the sub-query is evaluated only once. It however requires a table join for each word specified, therefore the maximum number of words that can be searched for is the maximum number of tables in a join, allowed by the specific DBMS, minus one to account for the series table itself.

14. The solution (5) uses one non-correlated sub-query for each word specified. The sub-queries are simply AND-ed together and there is no maximum to the number of words that can be searched for. Since the queries are non-correlated we may expect good performance from this solution.

15. In conclusion, the technique described above is applicable to any relational table with a textual or description column such as reference tables and footnotes tables. The technique is easy to implement and is particularly well suited to Internet and Intranet setting.

16. An important key to a successful Internet site is the search and navigation tools provided. Internet users are impatient with hierarchical search and navigation systems and even if persistent may quickly get lost in them. And of course you must never assume on Internet that a user will intuitively or even purposely select the same words as a producer and with the same meanings in looking for information. Statistical databases are no exception and may even be more frustrating to users because of their use of specialized, restricted vocabularies.

---

[5] These alternative solutions were proposed by Mr. B. Dragovic and Mr. D. Georgievski, consultants at the Statistics Division.

17. The approach described here is one strategy, which can be combined with others to offer Internet database users an "intelligent basket" of search tools. It is "free form" and text based and has two important strategic advantages. First, it easily adapts itself to any expansion of the database since it can readily be programmed to search any parts of the site or database whatever its rate of expansion. Second, it is easy to combine with various "translation tools" such as key words, synonyms, foreign language versions and so on, such that a search for "voitures" can equally hit on "automobiles" or "motor vehicles".

18. No single Internet search and navigation system for statistics will satisfy all user demands but the present version proposes one strategy, which combines an "open door" policy with behind the scenes programming to steer users in the most fruitful directions.

Annex

Soundex distribution

1. A soundex word index was created on the series or infotype table. At the time of the experiment the table contained 154 rows and each description had a maximum length of 160 characters. The resulting word index table contains 2191 rows but only 261 different (or distinct) word. The 261 different words produce 205 different soundex codes.

2. The results of the soundex distribution are summarized in the table below. It appears that with the exception of some words frequently used in the English language words (mainly prepositions and conjunctions) and words often used in the subject matter area being described such as "UN", "Division", "Accounts" "Prices" etc. the method has good discrimination. After about 20 "frequently" used words, which have occurrence counts between 20 and 150, the occurrence count drops to teens and then quickly to below 10. This means that most of the words, when used as search arguments, will return quite a small number of series or infotypes.

Soundex code	Number of occurences	Words
U500	145	the word "un" (for "United Nations")
D200	143	desa (our department's acronym)
S300	121	sd (for "Statistics Division")
A200	101	acc (for account), as, and surprisingly "age(17"
B000	100	the preposition "by"
N300	91	the words "nat" (for "national") and "net"
C536	72	central, country, countries and comtrad
P622	67	price, prices and projections
0000	67	numbers as in "1997", "5 years" etc.
C600	66	curr and currr
O100	63	the preposition "of"
S500	58	sna (from "System of National Accounts)
D300	50	data and death (death to the data?)
C514	48	the word "compiled"
I500	47	the preposition "in"
M100	37	mb (as in "Monthly Bulletin of Statistics")
E235	36	estimate and estimates
A530	28	the conjunction "and"
C523	26	const, constant and (surprise!) construction
P632	25	producing and production
C653	22	current
D100	21	div (for "division)
I532	21	index, indexes industr and industrial
V400	21	value
I516	19	import and imports
G320	18	goods
T000	17	"the" and "to" (surprisingly low number of occurences)
E216	15	exports and expressed
P143	15	population
C525	14	consumer, consumption

M362	14	meters and metric
N354	14	national
P600	14	per and pr
B652	13	branch
C300	13	city and code
C533	13	commodity
F650	13	from
I220	13	isic