SSC-N-688

# A Proposal for a Parameters Database

*O. Orban*

SSC Laboratory
Dallas, Texas

*ABSTRACT*

This paper describes a database for the maintenance of major SSC systems' parameters. The main purpose of the database and other functions it may serve are discussed, an implementation of the database is proposed, and possible screen interfaces and reports are described. Details of the database tables' structures and their relationships (the conceptual schema) are presented in an appendix.

December 15, 1989

# A Proposal for a Parameters Database

*O. Orban*

SSC Laboratory
Dallas, Texas

## Introduction

Since the time of the CDR, the idea of a parameters database has been discussed. The database was to aid in storing and updating the many tables of fundamental design parameters presented in the CDR, essentially creating an electronic version of the CDR's Attachment A which could easily be modified and printed. Thus, a current version of the parameters (and hence the SSC itself) could be referenced by various users. It was to be a 'controlled environment' in that, once in the database, a parameter's value could not be changed without a specific, recorded determination that the value should be y rather than x, e.g. there should be five dipoles per half-cell rather than six.

With the explosion in the number of tables and the amount of technical information in the SCDR, the idea of a parameters database has stopped being a need and has become an imperative. By providing a current, authoritative reference of high level parameters, a central clearing house for physics parameters would facilitate communications between design groups. Reports produced from the database would provide a source of accurate, consistent information for communication with or presentations to groups outside the laboratory. Finally, implementation of the database would allow managers and group leaders to review the parameters of various major SSC systems without interrupting work in progress at the various groups.

This paper presents a design for a database intended to contain and maintain a varied selection of SSC physics parameters. Many of the ideas that have been discussed in the past are incorporated into the design presented here.

## 1. Purpose of the Database

In creating a parameters database, the goal is to gather enough information into the database so that the major SSC systems are defined. Although the existance of a parameters database may be justified by its bookkeeping potential alone, the impetus for creating the database was derived from the idea of facilitating communications between design groups. It is hoped physicists and engineers would use the database as a reference to check data used in calculations and criteria used for design. To this effect, several types of paper reports and screen interfaces (forms) will be provided to access the information in the database. The reports must be layed out in a fashion similar to the SCDR tables used for input information so that the reports will be familiar and comprehensible to those who generate the stored information. The forms' layouts should be similar to reports' layouts, but the similarity will be subject to screen

constraints. Also, both processing in the forms and navigation between forms should be simplified by the use of mouse capabilities, where possible, and function keys. Finally, certain users will have the ability to suggest new parameters to add to the database, changes to existing parameters' values, and new catalogue headings. This function should also be performed by processing called from a form.

It should be stressed that the existance of the database is not meant to dampen inter-group 'what if?' discussions or 'what about?' technical notes. At the same time, the database is not meant to hold data needed to decide whether major parameters should be changed. (Such working parameters should be, as they are now, kept in private spread-sheets, databases, or papers.) It is meant to record modifications to the parameters, once they have been agreed upon by some means, in order to maintain an accurate design report of the machine.

## 2. Implementation of the Database

This section outlines the structure of and relations among database tables which would meet the users' needs. For more detailed information on the tables, see the appendix, which provides the details of the information to be kept in the tables.

### 2.1. the Tables and their Relationships

For simplicity, the database should consist of six types of tables:

a) a CATALOGUE table, maintaining the main relational key for the database

b) the parameter data tables, containing the parameters' names and values

c) the units tables, containing the units information associated with the various parameters

d) a table relating various parameters which must or may change if the value of one of the parameters in the relationship changes

e) tables containing suggested additions or changes to the parameter values or the catalogue system

f) a table containing the older parameter values which have been replaced by newer ones

Figure A at the end of the appendix provides a graphical sketch of the major relations between tables. Only the fields which relate the tables are shown in the figure.

### 2.2. an Overview

This section provides a brief overview of the database tables and the relations between them. It is intended to sketch a picture of the database as the user should see it. The appendix provides a tediously thorough discussion of the database as a programmer should understand it. First, a distinction must be made between a table in the database and a parameters table on paper, as in the SCDR. A table in the database will be called a table; a table on paper a report.

Essentially, the CATALOGUE table holds the title of a report and the report's relational key, a unique identifier assigned to a report based on its catalogue number and used in the database processing to identify the elements of other tables which

belong to the specific report. (To get a concrete idea of how the relational key functions, see the appendix. Note that it appears in all tables of the database.) The parameter tables hold the row headings and parameter values presented in a report. The values for one report are kept in a single parameter table; there is no cross-referencing among the several parameters tables. The units tables contain the units information of parameter values either as column headings or as supplementary row information. A report of current information is generated by using a report's key to collect and format the various informations from these three types of tables. Very simple.

The old parameter and suggested parameter tables serve as holding areas. The former table holds the previous values of parameters which have been changed to new values in the current parameter tables. This information is kept mainly to trace the past decisions. The SUGGESTIONS tables are provided so that users may propose changes to the database without over-writing the current contents of the database. The suggested parameter table contains proposed changes to the values of parameters in the current parameter tables. The suggested catalogue table contains proposed additions to the catalogue numbering system, and so expansions of the material in the database. Note that in Figure A there are no arrows leading to or from the SUGGESTIONS tables. This is to emphasize that the only link between them and the other tables is through user deliberations.

Finally, the user need not worry about the table of parameter relationships. At least, not yet.

### 2.3. a Few Words on the Catalogue Key

It is hoped that the numbering system used to reference parameter tables can be designed to reflect the relationships between system performance requirements and system design specifications chosen to meet those requirements. This will be attempted using a hierarchical approach in which a higher tier object contains the performance objectives or requirements of the systems listed immeadiately below. Conversely, lower tier objects are defined to contain system design choices which implement the preceding tier's requirements. As an example, the parameter list for the Linac will contain such items as linac emittance, while in the numbering system directly below Linac will be the parameters describing the ion source, the drift-tube linac, and other systems of the Linac. These lower tier parameters then describe the system choices made to attain the Linac operating specifications.

It is realized that the 'performance requirements vs. system design' issue is not a strictly hierarchic problem. For instance, even though the LEB and the Linac will appear on the same tier and have the catalogue prefix, e.g. 1-2, it is clear that the LEB injection beam requirements directly drive the Linac's output beam characteristics. However, this hierarchical approach will help to refine the problem and, if successful, limit inter-relations between systems to those which are on the same catalogue tier and have the same catalogue key prefix, e.g. such as 1-2-6 for all Linac systems.

This catalogue numbering system will be designed to perhaps the fourth or fifth tier, including only systems and sub-systems. It is meant to allow for the physics parameters which define the main systems and sub-systems, not to track the detailed information needed for a component specification document. Of course, the parameter database will contain information of interest to engineers. It is thought that the

catalogue key may parallel the higher levels of a WBS or technical component breakdown so that a user could trace the physics requirements which lead to specific engineering choices.

## 3. Interfaces to the Database

### 3.1. the Reports

The generic layout for current parameter reports is shown in Figure 1. A report will always contain the day and time of printing, appropriate column headings, and the catalogue number and report title of the parameters. Also, the page footer should consist of at least the page number and perhaps repeat the catalogue number. The number of Value fields will vary by the type of parameter in the report. If there are more than two parameter values, the Reference field will be dropped from the report and will then be available in other reports which list only the parameters' names and references.

A standard report from the database will contain the parameters which pertain to only one catalogue number, as shown by the Test Beams example. Other reports generated would have a similar format, but contain specialized information selected from different catalogue numbers. For instance, a report could be produced which contained all level 2 technical objectives for the injector systems. This type of report would list each of the parameters under their respective catalogue numbers and titles, for reference to the system. Additional reports would print the information in either of the SUGGESTIONS tables and a history of changes made to specific parameters.

### 3.2. the Screens and System Processing

Using Sybase's forms-oriented language, APT-SQL, it is anticipated that all processing which users routinely need can be supplied by APT forms. Such processing will include looking up parameter values on screen, requesting reports, and suggesting changes to parameter values. Figure 2(a) shows a typical APT form which is divided into two parts - one containing the form title and the processing menus and the other displaying the information from the database tables. Note that Figure 2(b) shows functions available under the 'Tiers' menu. Those menu items without an '!' in the title have function selections hidden beneath them; those with an '!' call a processing routine directly.

The interface may require four distinct types of forms:

a) a Catalogue form

b) various Parameter forms

c) two Suggestion forms

d) several Help forms

The Catalogue form serves as a navigational aide which allows the user access to the specific set of parameters for which they wish information. In its data area, the form will display CATALOGUE table data which are on the same tier and have the same catalogue prefix. In the menu area of the form, it will display menu items which allow the user to move between catalogue tiers, display the parameters of a catalogue entry, print a report, and, for those users with appropriate permission, suggest new entries. The form may look something like the form shown in Figure 2(a). Figures

| | Parameter Description | Value | Units | References & Remarks |
|---|---|---|---|---|
| 3.5 | TEST BEAM PARAMETERS | | | |
| .5 .1 | HEB Beam Line | | | |
| | primary beam energy | 2000 | GeV | |
| | primary intensity | 1.0 E14 | protons / pulse | |
| | cycle time | 2.5 | min | |
| | spill length | 30 | sec | |
| | scheduled operating time | 175 | days/yr | |
| | cycles per yr | 1.0 E5 | per yr | |
| .5 .2 | MEB Test Beam Lines | | | |
| | primary beam energy | 200 | GeV | |
| | primary intensity | 1.0 E14 | Protons / pulse | |
| | cycle time | 3 | sec | |
| | cycles per hour for test beams | 24 | | |
| | spill length | 1 | sec | |
| | schedules operating time | 175 | days/yr | |
| | cycles per year | 1.0 E5 | | |
| .5 .3 | Secondary beam (wide band beam optics) | | | |
| | maximum energy transmitted to calibration hall | 2000 | GeV | |
| | maximum secondary beam energy | 1750 | GeV | |
| | minimum secondary beam energy | 1 | GeV | |
| | lepton identification | 1000:1 | | |
| | hadron identification | 100:1 | | |
| | absolute energy resolution | 0.33 | % | |
| | relative energy resolution | 0.05 | % | |
| | particle flux above 100 GeV | > 1.0 E2 | particles/sec | |
| | particle flux at 2 TeV at calibration hall | 10 E7 | protons/sec | |
| | muon background at calibration hall | < 150 | kHz/m^2 | |
| .5 .4 | Switchyard transport magnets | | | |
| | no. of electrostatic septa | 30 | | |
| | no. of 2-way Lambertsons | 6 | | |
| | no. of 3-way Lambertsons | 12 | | |
| | no. of 4 Tesla dipoles | 66 | | |
| | no. 3Q120 quadrupoles | 121 | | modified from SSC-230 by R.Coleman on 9/14/89 |

End of Listings

Figure 1

2(b) - 2(d) display some possible function options.

There will be two ways to access the parameters. For users unfamiliar with the catalogue numbering system, it will be possible to find the catalogue entry of interest by working down the hierarchical system from the first tier. With the mouse, the user would select one from among the first tier entries, which will automatically be displayed when the program is started. Then, selecting the 'Down One Tier' from under [Tier] , the form would display the catalogue entries below the selected entry. Working down the catalogue tree in this way, he can find the proper catalogue entry, select it, and then use the 'Display For Selected' item under [Parameters] to display the parameters pertaining to the entry. Thus, if a user selected Coupled-Cell Linac in the Catalogue form and called for its parameters, a form like the one in Figure 3 would appear. Alternately, if the user knows the entry's catalogue number or title, he can, by using one of the Find selections under [Parameters] , directly access those same parameters.

The data section of the parameter forms would have a look similar to the generic report of Figure 1, with variations imposed by screen limitations. Whereas there is a line break between the parameters' headings on the paper, only one heading's parameters will appear on the screen at a time and provided functions might be used to move to the next or previous heading. Also, because all the parameters in one heading may not fit on a screen, the scrolling arrows, seen shaded in Figure 3, would be used to view those off the screen. Because of the screen width, the Reference field would not appear on the standard parameter screen. In the menu area of the form, selections will allow all users to find a specific heading by its number or title, to print a report of the parameters, and to view the parameters and their references on another form which does not contain the parameters' values. Figure 3(a) shows a form for displaying one-valued parameters; figure 3(b) shows the same form with functions options under 'Heading' displayed.

If the user has permission to suggest changes, a suggestion menu will appear in the menu list for both the Catalogue and the Parameters forms. By selecting [Suggest] in either form, an appropriate suggestion form will over-lay the window. In both cases, the data area will provide enough fields to accept all information necessary to define a parameter or a catalogue entry. The processing menus provided will allow a user to find entries already in the suggestion tables, to modify an existing suggestion only if the user suggested it, and to add new suggestion to the tables.

Finally, a good number of pop-up help windows can be attached to each of the three types of forms previously described. By selecting the [?] icon in a window, the user may call up any help attached to the form.

It must also be stated that the fore-going discussion applies to users with SUN systems. Similarly functioning forms may be produced for machines able to emulate a VT100 session under UNIX. However, the selections and other processing which a SUN user would be able to perform with a mouse must be mapped to the function keys and control keys of the other machines. Thus, a user with a Mac, an IBM PC or a PC clone *should* be able to access the forms and view the parameters.

## Conclusion

This paper has sketched an implementation of a database meant to hold and update the physics parameters of the SSC. The implementation presented is workable, and some programming for it has already been done. However, many points have not yet been decided. The catalogue system has not yet been delineated and some of the table layouts may change. For instance, it has been suggested that, for simplicity, there be only one parameter table and it contain several Value_n fields. Those parameters which need less than n values to specify them would have null values in those Value_n fields not used. With only one parameter table, the layout would appear simpler and the processing would lend itself more readily to APT-SQL, because there would be more do-loops and fewer if-branches involved. However there would be some (arguably small) cost in storage memory space and, perhaps, retrieval time.

It is seen then that this paper represents work in progress, and useful suggestions generated in response to this paper may be incorporated into the database design.

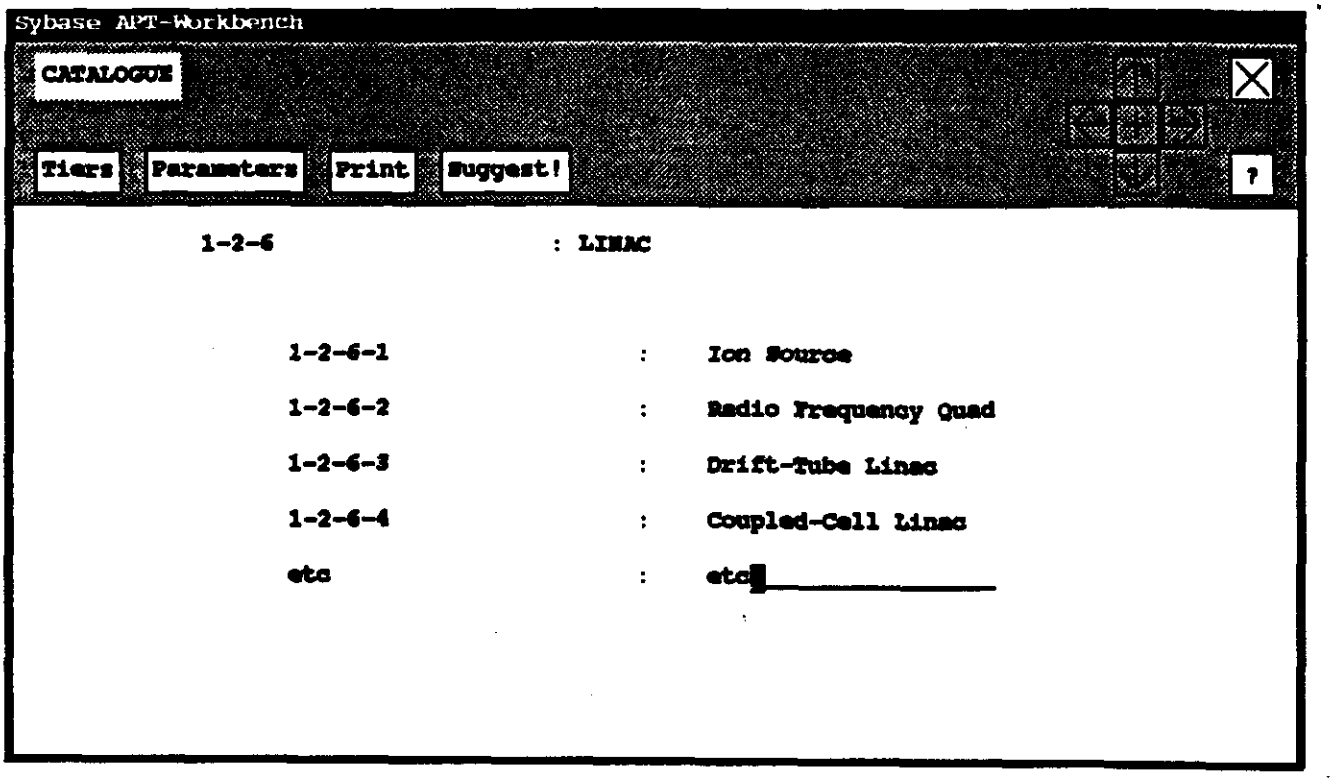


Fig. 2(a) A Possible Layout for the Catalogue Form

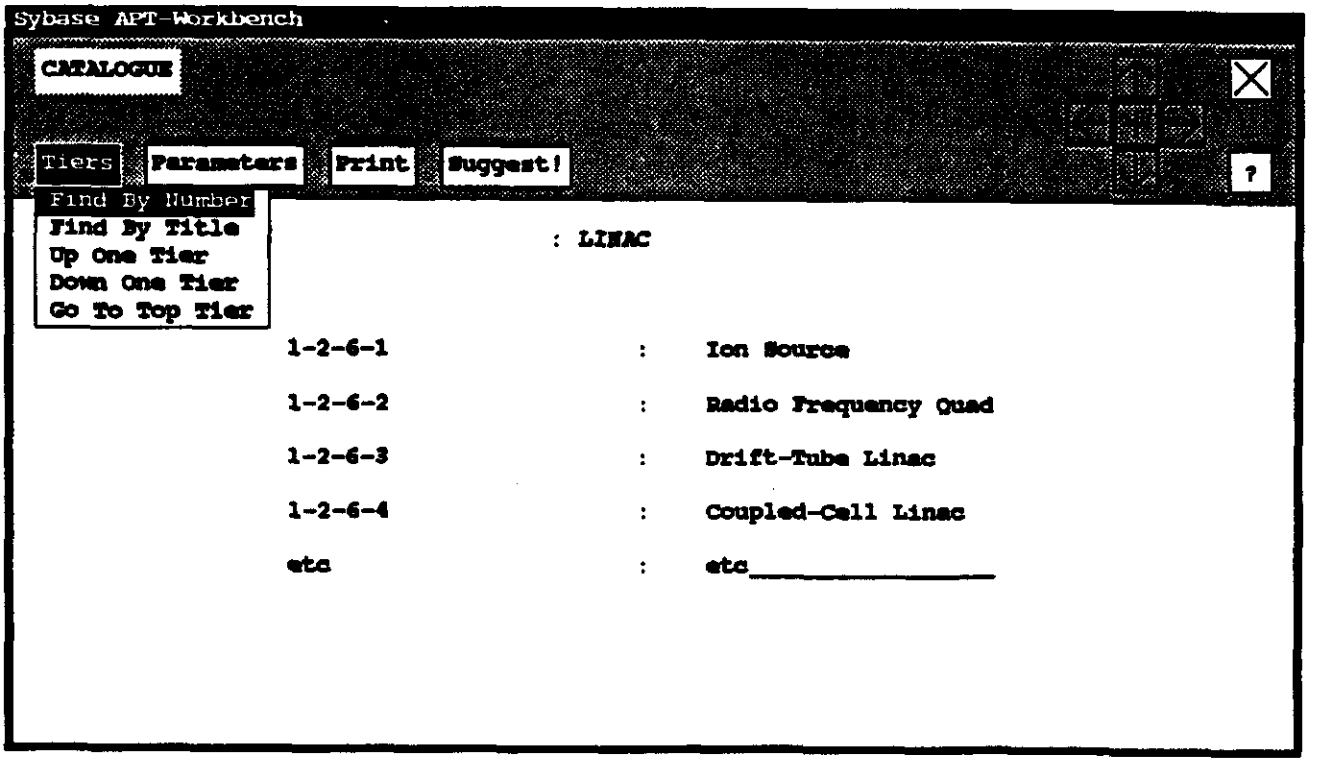


Fig. 2(b) Catalogue Form showing the Functions available in the 'Tiers' Menu

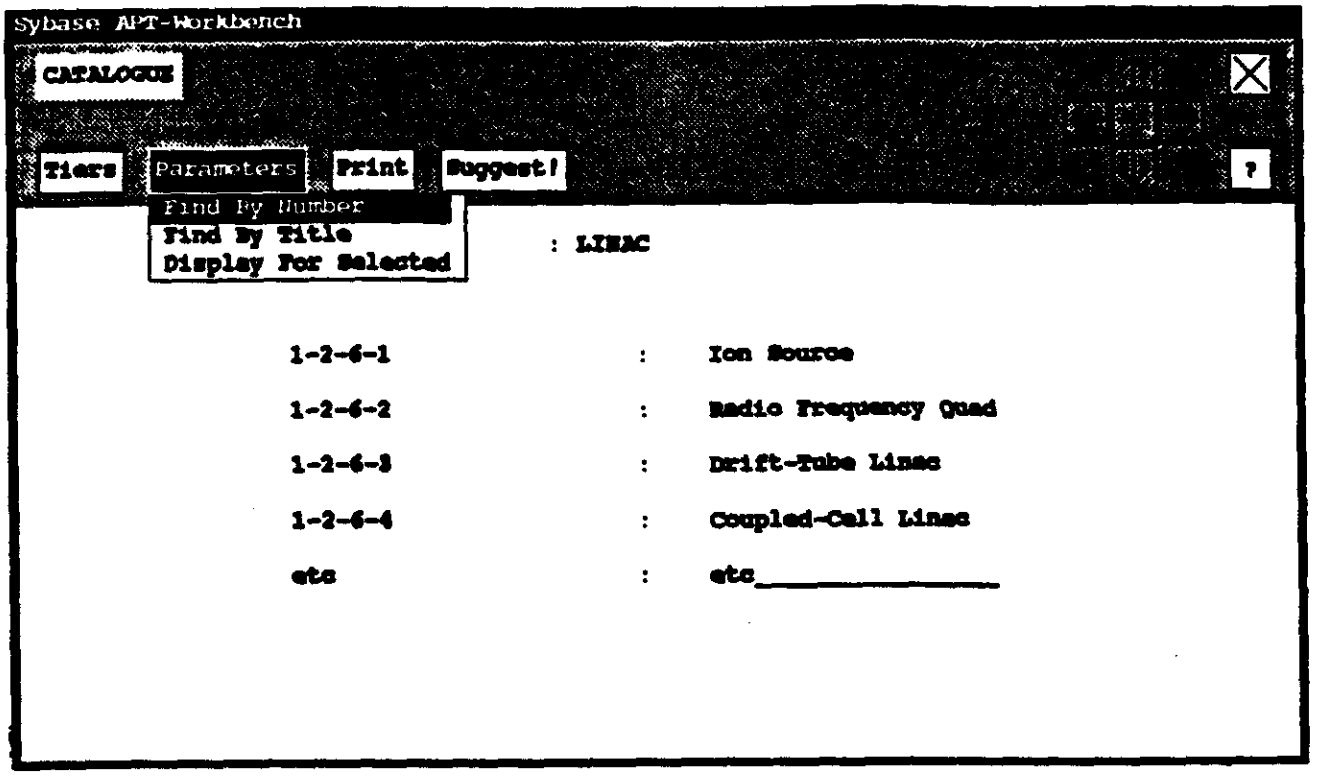


Fig. 2(c) Catalogue Form showing the Functions available in the 'Parameters' Menu

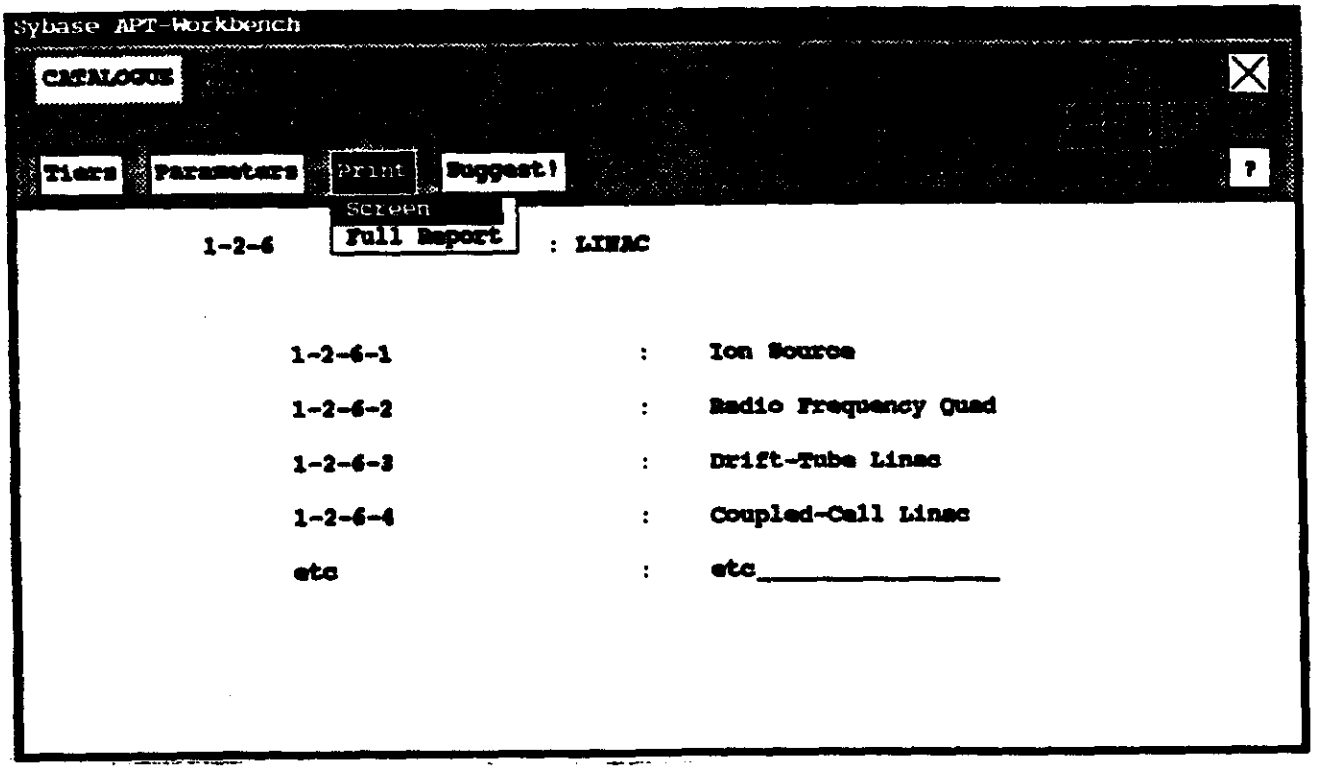


Fig. 2(d) Catalogue Form showing the Functions available in the 'Print' Menu

Sybase APT-Workbench

PARAMETERS

Headings | References! | Print | Suggest! | ?

1-2-6-4 : Coupled_Cell Linac

| | | |
|---|---|---|
| CCL Type | side-coupled | |
| frequency | 1284 | MHz |
| output energy | 600 | MeV |
| pulse length | < 36 | sec^-6 |
| trans. emittance (n,rms) | 0.15pi | mm-rad |
| | etc. | |

Fig. 3(a) A Possible Layout for a Parameters Form

Sybase APT-Workbench

PARAMETERS

Headings | References! | Print | Suggest! | ?

Find
Next
Previous

4 : Coupled_Cell Linac

| | | |
|---|---|---|
| CCL Type | side-coupled | |
| frequency | 1284 | MHz |
| output energy | 600 | MeV |
| pulse length | < 36 | sec^-6 |
| trans. emittance (n,rms) | 0.15pi | mm-rad |
| | etc. | |

Fig. 3(b) Parameters Form showing the Functions supported in the 'Headings' Menu

## APPENDIX

### The Tedious Details of the Parameters Database

This appendix contains database table layouts, which specify fields and their datatypes, and a relational map of the database (figure A), which shows the relations between tables and the fields used as keys in these relations. Note that this design is a working design, and so, subject to change as other processing needs or faster processing schemes present themselves.

### 4. the CATALOGUE table

This table maintains the main relational key used throughout the database. The entities which its rows describe are the elements of the report catalogue system. These elements are represented, to the user, by a report title, and, in the database, by a unique relational key which is a number in the catalogue system. The catalogue numbering system will divide the parameters by the SSC systems and sub-systems which the parameters define and will probably parallel the higher levels of the work breakdown structure (WBS) or a technical component breakdown. (For a more detailed discussion of the numbering system, see 'a Few Words on the Catalogue Key' in the main body of this report.) Other information contained in this table will include the name of a person repsonsible for suggesting changes to the parameters associated with the key and for discussing the effect that changes to other parameters might have on the parameters in his/her area. In addition, several pages of descriptive text may be included, if necessary, to provide an introduction to the system the element represents, i.e. basic definitions, variables, and the function the system performs.

All user interfaces and processing calls will reference the CATALOGUE table, so it is imperative to maintain the integrity of the key in this table. Trigger level SQL will ensure that if a change is made to an existing key, this change will then ripple to the other tables. Also, SQL programming will help provide trivial, but necessary, features such as inserting a new key into the middle of a set of uniquely indexed keys, checking that a catalogue entry exists before a first parameter for the key is accepted, and checking that no parameters are listed for an entry before deleting it.

This table consists of the following fields and field types:

| CATALOGUE | |
|---|---|
| Field | Datatype |
| volume | tinyint |
| chapter | tinyint |
| section | tinyint |
| paragraph | tinyint |
| catalogue_# | char(16) |
| table_# | tinyint |
| row_col | logical |
| title | varchar(50) |
| responsible | varchar(50) |
| description | text |

- Volume, Chapter, Section, and Paragraph are the primary keys for the table and are indexed. Upon insert or update of a row, a catalogue number is automatically produced from these four fields by provided SQL processing.
- Catalogue_# is the foreign key of the table which links to all other tables. It contains the catalogue number which will be maintained by trigger level SQL.
- Table_# is used to determine the parameter and units tables to be called for processing.
- Row_col determines the type of units table to be used in processing. T = row; F = column. (see 'the UNITS tables')
- Catalogue_# and Title should always appear across the top of a report or screen.
- Responsible contains the name of the group leader, or his/her designate, who has the responsibility of maintaining the data in the section and proposing changes to the data.
- Description is a text field which is meant to hold any text which explains the system and its parameters.

## 5. the PARAMETERS tables

There will be several parameter tables. Some parameters may only be properly specified with several values; the value the parameter takes depends upon the section of the machine or the mode of operation under discussion. For example, parameters which specify the r-f ramp profile may take one of four values dependent upon the mode and energy of operation (storing v. ramping, 2 TeV v. 20 TeV). Thus, the several types of parameter tables allow all types of parameters to be stored and processed in a systematic fashion. Initially, the greatest number of allowed values per parameter may be six (this depends on an examination of the SCDR). The system, however, may be easily expanded to handle parameters which need, say, ten values for full specification. Information in the CATALOGUE table will specify which parameter table holds the data for a specific catalogue number.

The data in a parameter table will include the parameter's name, its catalogue number, its value or values, and a reference field for comments, such as a fuller definition of the parameter, an equation from which the parameter is derived, or a citation of such publications as the SCDR or SSC technical notes. Other information which facilitates internal processing will also be maintained. Finally, information relevant to particular parameters will be maintained. For instance, if the catalogue numbering system should differ from the WBS, then the technical objective level of a parameter may be maintained in a seperate field called Level. Even though the HEB peak output energy would appear under the catalogue title of HEB, which may have a catalogue number of 1-2-2, this parameter would be identified as a level 2 project technical objective by placing a '2' in the Level field. The Level field then allows special reports to be generated which contain only level 1, 2, or 3 parameters. (For more on technical objective levels, refer to the Draft Project Management Plan, revision IV, or wait for the movie.)

The naming convention for the Parameters_n tables is that the 'n' number equals, rather obviously, the number of Value fields in a parameter table, which, in turn, is derived from the number of values needed to specify a parameter fully. The structures of Parameters_1 and Parameters_2 are shown below. The other Parameters_n tables differ from Parameters_2 only in the number of Value and Change fields they have.

| PARAMETERS_1 | | PARAMETERS_2 | |
|---|---|---|---|
| Field | Datatype | Field | Datatype |
| catalogue_# | char(16) | catalogue_# | char(16) |
| heading | tinyint | heading | tinyint |
| line | tinyint | line | tinyint |
| object | varchar(150) | object | varchar(150) |
| value | varchar(50) | value_1 | varchar(50) |
| units | varchar(50) | change_1 | logical |
| change | logical | value_2 | varchar(50) |
| reference | varchar(150) | change_2 | logical |
| | | reference | varchar(150) |
| changed_at | timestamp | changed_at | timestamp |
| suggested_by | smallint | suggested_by | smallint |
| related | logical | related | logical |
| report | logical | report | logical |
| level | char(3) | level | char(3) |

- Catalogue_#, Heading, and Line serve as the primary key for the table and are clustered indexed. Catalogue_# also serves as a reference to the Catalogue table and the Units_n tables. Catalogue_#, Heading, and Line serve as reference to the Units_row table.

- Heading allows the parameters to be split by line breaks in reports and by page breaks on screen.
- Object, Value, and Units are self-explanatory. Reference contains any comments or citations (such as SCDR page numbers) that may be relevant.
- Change and Change_n fields show which values have had modifications suggested for them but not yet implemented. These values are then assumed to be under discussion, so this field serves as a warning flag that the value may change in the near future. If a Change_n is true, the value it applies to will have an '*' next to it in reports.
- Changed_at and Suggested_by monitor modifications by automatically recording the time a value was modified and the user id of the person who suggested the new value.
- Related provides a check as to whether the parameter is related to any other parameters. (see 'the RELATIONS table')
- Report allows specific parameters from within different catalogue headings to be pulled together into one report.
- Level contains a number identifying a parameter's technical objective level.

## 6. the UNITS tables

It is readily apparent that only the units of the single valued parameters are held in the parameter tables. For a multi-valued parameter, the units of the parameter's value may vary from value to value, i.e. changing for the different values used to specify the parameter. In this case, the table of paramter data is structured so that the units are constant down the table's columns. Alternatively, the parameter's values may all have the same units, so that the units are said to be constant across the row. An example of a report with the same units down its columns would be a report which characterized the various arc correction elements by a measure of magnet field strength (a line integral with units Tesla-meters), maximum current, volts, etc. An example of a report with constant units across the rows would be the ramp profile description which specifies different parameters (such as peak voltage and rms bunch length) in different operating modes (2 TeV storage, 2+ TeV ramping, 20- TeV ramping and 20 TeV storage).

Parameters of both types are allowed for in this database by providing several types of units tables. Only one table is necessary to contain the units for parameters which have constant units across the row, but the catalogue number plus a supplementary key are needed to link a specific parameter to its UNITS_ROW table entry (see appendix). To accommodate parameter reports which have constant units down the columns, there are as many units tables as there are multi-parameter tables. In this latter case, only the catalogue number is needed to relate a table of multi-value parameters to its units table. The information that determines which type of units data a report contains is held in the CATALOGUE table in the Row_Col field.

Only the structures of Units_Row and Units_2 are shown below. The other Units_n tables differ from Units_2 only in the number of Units fields they contain. A similar naming convention is in effect for the Units_n tables as for the Parameters_n tables.

| UNITS_ROW | | UNITS_2 | |
|---|---|---|---|
| Field | Datatype | Field | Datatype |
| catalogue_# | char(16) | catalogue_# | char(16) |
| heading | tinyint | | |
| line | tinyint | | |
| units | varchar(50) | units_1 | varchar(50) |
| | | units_2 | varchar(50) |

- Only the Catalogue_# is needed as a reference for those parameters with constant units across the rows. The Units_n tables are clustered indexed on this field.
- Catalogue_#, Heading, and Line are all needed as a primary key to link the different units with the specific parameter to which they pertain. The Units_row table is clustered indexed on these fields.

## 7. the RELATIONS table

The RELATIONS table is provided to allow two parameters in the main parameter table to be linked. The reasons to link two parameters vary. If the same parameter were to appear under more than one catalogue number, using the RELATIONS table, processing would be provided which automatically updates all occurances of the parameter if one occurance is modified. Another reason two parameters might be linked is that, because of physics or engineering considerations, a change in one parameter's value might necessitate a change in the other's value. It is expected that this table's capabilities will not be implemented immeadiately, unless parameter redundancy does occur.

| RELATIONS | |
|---|---|
| Field | Datatype |
| catalogue_1 | char(16) |
| heading_1 | tinyint |
| line_1 | tinyint |
| catalogue_2 | char(16) |
| heading_2 | tinyint |
| line_2 | tinyint |
| duplicate | logical |

- The order in which the two related parameters are listed is irrelevant.
- Duplicate allows for the possibility of data redundancy in the parameter tables. If Duplicate = .T., then any changes to one parameter in the relationship are copied to the other automatically. If Duplicate = .F., then this suggests that a change in one parameter would affect the other's value(s) in some way which involves a

discussion.

## 8. the SUGGESTIONS tables

Certain users will have the ability to suggest new values for existing parameters, new parameters, and new catalogue entries. However, they will not be allowed to directly affect the current parameters' values. Two tables will be provided to store these suggestions until they can be decided upon. The SUGGESTIONS_P table will contain a newly proposed value and all information necessary to relate it to a specific parameter value in the parameter tables. Also, this table will contain information which identifies who suggested the modification and when it was suggested. If the change is adopted, the user id of the suggester will automatically be stored with the new parameter in the main tables. The SUGGESTIONS_C will contain similar information for any additions or modifications to the catalogue system.

A user's ability to suggest changes will be limited to the system or sub-system the user is responsible for. One possible way to implement these limitations is by creating views of sections of the table data and granting insert permissions on the appropriate view only. A view is a logical database table which is created by specifying and recording SQL selects and joins on actual tables in the database. These recorded SQL statements then take on the attributes of real tables, so one may assign 'write' permissions to them.

| SUGGESTIONS_C | | SUGGESTIONS_P | |
|---|---|---|---|
| Field | Datatype | Field | Datatype |
| volume | tinyint | catalogue_# | char(16) |
| chapter | tinyint | heading | tinyint |
| section | tinyint | line | tinyint |
| paragraph | tinyint | | |
| table_# | tinyint | | |
| row_col | logical | | |
| title | varchar(50) | object | varchar(150) |
| responsible | varchar(50) | value_# | tinyint |
| description | text | value | varchar(50) |
| | | reference | varchar(150) |
| insert | logical | insert | logical |
| suggested_at | timestamp | suggested_at | timestamp |
| suggested_by | smallint | suggested_by | smallint |

- Catalogue_#, Heading, and Line link the SUGGESTIONS_P table to the various parameter tables. At this point, it is not envisioned that any fully automatic processing will link these tables with any of the others.

- Object (the parameter's name) may be modified at this time, but this change must be accompanied by a change in the parameter's values.
- Value_# contains the column number for a modified value of a multi-valued parameter.
- Insert indicates whether or not a suggested catalogue number or parameter is already included in the database and therefore whether it must be inserted into the appropriate table.
- Suggested_at and Suggested_by monitor the changes by automatically recording the time of the suggestion and the user id of the person who made the suggestion.

## 9. the OLD_PARAMETERS table

This table is meant to help track modifications of the main parameter tables. The old value of any modified parameter is automatically inserted into this table. The information in the table will be identical to the information in the SUGGESTIONS_P table except that the Reference field of a parameter will not be recorded. Also, the Suggested_by field will contain the name of the user who originally suggested the old value, not the user who suggested the value which replaced it. Note that this information is maintained solely for reporting purposes. Old versions of the main parameter tables must be re-created from tape back-ups.

| OLD_PARAMETERS | |
|---|---|
| Field | Datatype |
| catalogue_# | char(16) |
| heading | tinyint |
| line | tinyint |
| object | varchar(150) |
| value_# | tinyint |
| value | varchar(50) |
| updated_at | timestamp |
| suggested_by | smallint |

- See the Suggestions table above.
- Suggested_by will contain the user id of the person who originally suggested the old parameter value, not the person who suggested the latest modification.
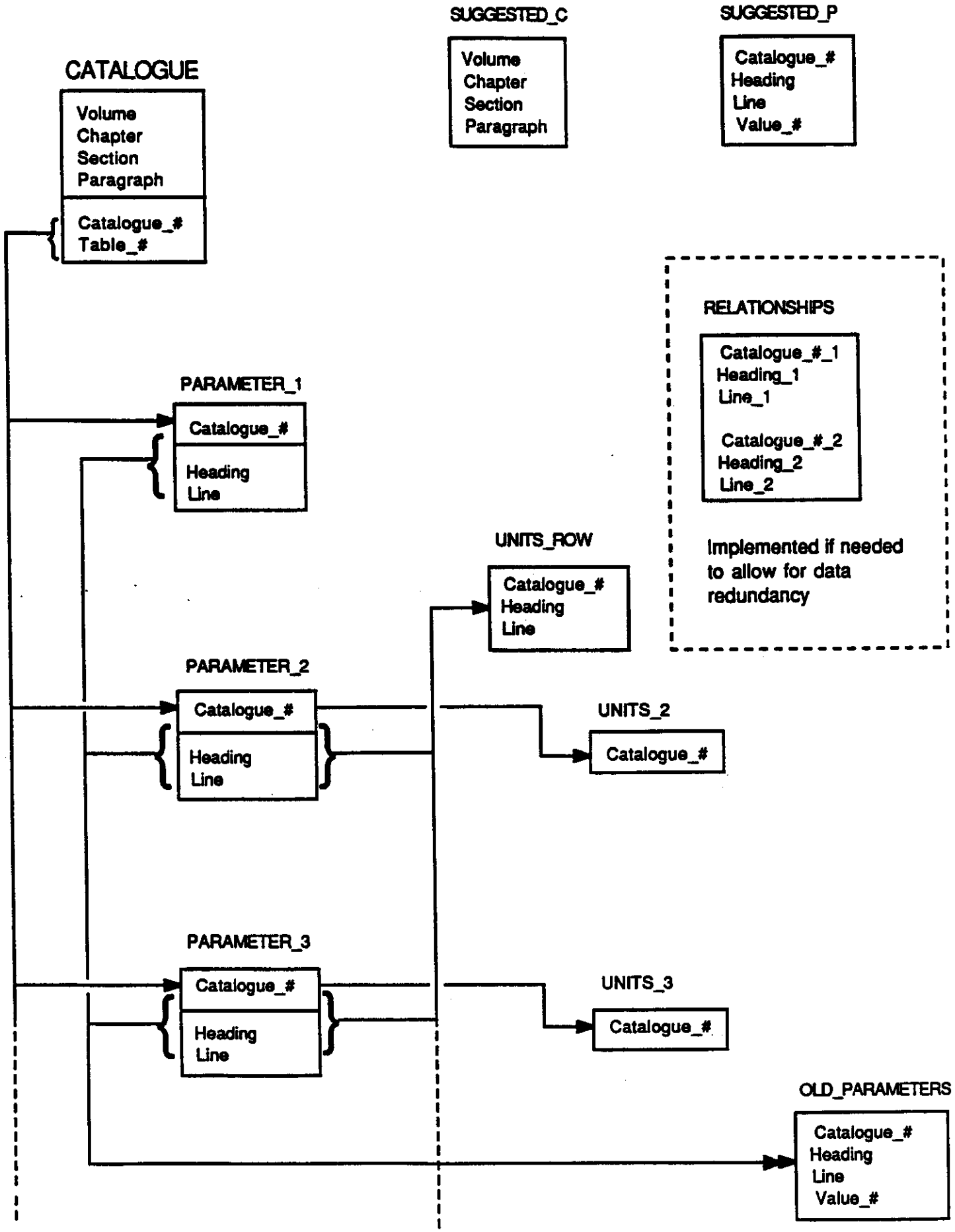
Fig. A Relations between Database Tables