SSC-N-582

General Remarks on the Organization and Contents of the Proposed Cable Data Base

T. Ferbel

SSC/CDG and The University of Rochester

(January 11, 1989)

This document contains the first version of the SSC/Cable data base structure, which is about to be implemented using the Sybase Database Management package on a SUN computer. The structure and contents reflect many discussions (and often compromises) among a variety of technical experts, scientists and engineers, all associated in one way or other with the SSC project. This work was initiated by a request from T. Kirk, Head of the SSC Magnet Division, to provide a reliable and user-friendly system that would enable one to correlate properties of any magnet with its structural raw materials. Kirk appointed a "Commission on Cable Data", with representatives from all laboratories involved in the SSC magnet R&D activity: J. Carson(FNAL), B. Leedy(CDG), R. Remsbottom(CDG), R. Scanlan(LBL), P. Wanderer(BNL), with T. Ferbel as chair. The initial phase was to aim at the organization of a general data base, using a standard nomenclature, that could serve the needs of all the laboratories, and trace the finished superconductor cable to its source contents. [Because of their great expertise with magnet data bases, P. Baggett(BNL) and B. Brown(FNAL), were immediately recruited into service on the Commission, and have contributed extensively to the generation of the following structure.]

An orientation meeting, held on October 5, 1988 at the SSC/CDG Headquarters at LBL, was the effective starting point for this activity. From the beginning, it was recognized that a relational database management system was required to provide flexibility for any future changes in the structure or philosophy of the retrieval and inventory of magnet data. Sybase was discussed extensively during the meeting, and because it was highly recommended by Kirk and the CDG, we have proceeded with this system in mind.

The Commission's first goal was to specify the variables thought important for tracking and maintaining the quality of the final product, and then to determine which of these variables could be elicited from vendors without impinging on any proprietory or otherwise sensitive information. In doing this, those of us who were not experts, had to develop a relatively detailed understanding of all the aspects of the production process, to assure that the data base we were setting up was not flawed ab initio.

The proposed database consists of various tables (equivalent to files), which contain variable names (columns) and the type of data (e.g., integer, floating point, character) to be entered in any given column. An indication is given when a NOT NULL entry will be required (i.e., an entry will have to be made), and the units, if any, to be used for that particular variable. A descripion of the contents of the column are also provided; this information will be contained in something akin to an alias table that will be available for look-up purposes. Data entries for each full set of variables will appear as rows in any table.

Several of the tables (e.g., COMPOSITE_BL_COMPONENT) are relatively complicated, and many of the id definitions are not completely transparent, and require some additional explanation for proper use. Appropriate comments will be provided in glossary tables, such as TABLE_COMMENTS and ID_FORMATS, to help users navigate more easily through the system.

In designing the data base we have attempted to set up a system that uniquely specifies and inventories any sample or spool of wire and/or cable that is produced for the SSC. Most data will be entered only once into only one column of a table, but it may appear for convenience (through look-up possibility) in different places. There will, of course, be important exceptions to this rule, because we may wish to connect a particular strand id number, for example, to a purchase order, and this necessitates multiple entry of one of these items. In fact, ID's in general will tend to appear in several tables in order to connect any item to its many properties. The more complicated ID's will be formed by concatenation of the simpler individual bits of entered information.

We have assumed that, once full-scale production commences, data entries will be made through electronic means. It is essential that some kind of access be provided to the data base from the manufacturers' assembly lines. This will be particularly important if it turns out to be necessary to keep track of cold welds in the spools of wire (strand maps of the cables). Purchase orders will specify the cable serial numbers that the vendors are to use (CABLE_PROD_ID). Reels will be marked with the SSC ID before they are delivered. Each spool of cable will have on its exterior packaging, as well as on the spool itself, the information required for forming the CABLE_SPEC_ID and the CABLE_PROD_ID, to uniquely label the nature of the cable and its serial number at production. Whenever cable is to be cut, a sequential cut-number for any original spool will be recorded and kept in the data base (as well as on the spool of the cut-off piece). In addition, a cable segment id (CB_SEG) will be added by the user as the spool is cut up, and as parts are respooled. Some variety of bar-code markings will clearly be required once the system is in full operation. It is also understood that there will be facility for respooling cable at any location where cutting will be required.

Each spool of strand will also have on its exterior packaging, as well as on the spool itself, the information to form the STRAND_SPEC_ID and the ST_PROD_ID. As in the case of cable, when wire is used for other than normal cable production, a ST_SEG id will be added by the user as the spool is cut up, and as parts are respooled.

It might be appropriate to highlight at this time some of the decisions we have made concerning the structure of the proposed data base. We have selected key features of strand and cable, and used these to form specification id's that can be used to label and to trace any item to its spec documents. For example, the STRAND_SPEC_ID or CABLE_SPEC_ID fully describe the properties of the different types of SSC strand or cable, and point to the documents in which they are specified. The INSULATION_CODE for cable is another id of this type, but one that has yet to be fully defined (spec documents on this do not exist as yet). We feel that the data base has sufficient flexibility so that if the character of the cable, the strand, or any processing, should change in the future, the new procedures will be able to be incorporated in a reasonably straightforward fashion.

We have chosen to use vendor-provided id's for tracking superconductor ingots and copper material (heat numbers and lot numbers) as well as all spools of strand. (Nevertheless, we have also left the possibility for using a parallel numbering scheme for billets, should we find this desireable.) For any cable, or for special strand (e.g., for use in correction coils, etc.), we will have our own serial numbering system that will require vendor compliance.

Because it is difficult, even for physicists, to completely foresee the future, we have made the production tables (e.g., the COMPOSITE_BL_TABLE) quite general. This, again, should provide us the desired flexibility, once changes in present procedures are eventually required. However, we wish to make it absolutely clear that we have decided that it will suffice, at least for routine purposes, to track the contents of any item, for example a billet, in only a rather qualitative manner. From the information in the data base we will know which batches of material were used in the formation of any billet, and will have the vendor's assurance that each of these elements was of the appropriate quality (via a certificate of conformance), but we will not know the quantitative makeup of any stack. Nevertheless, should more detailed information be required, we

can obtain private access to the vendor's records on these matters (if such details are, in fact, available!).

Again, because of uncertainties in final production methods, we have introduced tables that are just skeletal in nature, and which can be expanded or eliminated as needs arise. (CABLE_COATING and STRAND_INSULATION are two such tables.)

To simplify inventory, we have separated production information from receiving and use information. Thus inventory of normal strand appears in two tables. Production and use of cable (or special strand) is more complex, and we have separated that information into three tables. Any produced cable is recorded in CABLE_PRODUCTION, its usage is tracked in CB_PROD_INVENTORY, and after it is wrapped, it is tracked (along with special strand) in SPOOL_INVENTORY_USE. Thus, starting with any piece of cable in the last table, we can trace it back through the data base to all its contents and production routes, and find out whether there are is any more similar or identical cable available somewhere in storage. (This is, of course, at this time, still just in principle!)

Before proceeding with the coil and magnet sections of the data base, we plan to decide on the kind of standard reports that might be useful for assessing the quality of cable and strand material. We also intend to transfer all currently stored data to the Sybase system, to establish some preliminary interactive procedures for data entry, transfer and interrogation, and, of course, to use the system in order to decide whether it is in tune with the needs of the SSC.

Finally, we have attempted to standardize the nomenclature, not only for data on strand and cable, but also for coil and magnet description. To this end we have now chosen a preliminary scheme which, hopefully, will satisfy all elements of our community, and be adopted in the near future.

****
* Following is latest version of the magnet data base for SSC strand & cable *
****

For ease of reference, the table descriptions have been grouped into sections, and each table has been given a number. The sections are as follows:

```
section  1 = vendor, purchasing, and general information
         2 = specifications for strand and cable
         3 = ingot & billet production & tests; strand production & receiving
         4 = strand tests
         5 = cable production & receiving
         6 = cable tests
         7 = coil information
         8 = cold mass assembly information
         9 = cryostat assembly
        10 = magnet data
        11 = comments on tables, formats & naming conventions
```

The presence of a pound sign (#) in any first column indicates that the variable ("field" or "column" in any table) listed above it is not allowed to have a null entry. Whenever appropriate, units for any variable appear between "less than" signs (<units<) following the #, or in that first column in place of the #, if null entries are permitted. If no # and no <units< appear, this means that the entries can be null, and there are no units for the variable in question.

For those who are not Sybase enthusiasts, the data types listed below have the following meanings:

```
int = integers (+/-) up to 2**31 (four bytes)
smallint = integers (+/-) up to 2**15 (two bytes)
tinyint = integers (+/-) up to 2**8 (one byte)
```

float = floating point (eight bytes)
char(n) = fixed length (n = 1,...,255) of characters (any kind: T6%#43Q)
varchar(n) = variable length (up to n) of characters (any kind)
datetime = date, with lots of formats possible (we use "mm/dd/yy")

The last three types are all entered within quotation marks (single or double)

.....
.....

1.0 VENDORS

VENDOR_ID_MNEM	varchar(8)	mnemonic for manufacturer or supplier (e.g., BNL,IGC,LBL,FUR,SUN)
#		
VENDOR_ID_CHAR	char(1)	one character id to be used as prefix on various items supplied by vendor; obtained via look-up from ORGANIZATION_MNEMS table
#		
VENDOR_NAME	varchar(50)	name of supplier
#		
VENDOR_DEPARTMENT	varchar(50)	appropriate department
VENDOR_STREET	varchar(50)	first part of supplier's address (street name and number)
VENDOR_BUILDING	varchar(50)	second part of supplier's address (e.g., name of building)
VENDOR_ADDRESS_ETC	varchar(50)	third part of supplier's address (other)
VENDOR_CITY	varchar(20)	city of supplier
VENDOR_STATE	varchar(20)	state of supplier; standard US mail TWO character abbreviation preferred
VENDOR_ZIP	varchar(10)	supplier zip code (left adjusted, with hyphen and subcode, when available)
VENDOR_COUNTRY	varchar(20)	supplier country

.....

1.1 VENDOR_PRODUCTS

VENDOR_ID_MNEM	varchar(8)	supplier mnemonic
#		
PRODUCT_MNEM	varchar(20)	kind of product available from supplier (e.g., SC_INGOT,SSC_KAPTON,SC_CABLE,CU_CAN, SC_WIRE, etc)
#		

.....

1.2 ORGANIZATION_MNEMS

==========

ORGANIZATION_MNEM	varchar(8)	mnemonic for any laboratory or vendor (e.g., FNAL, RHIC, TELEDYNE, IGC, SSC)
#		
ORGANIZATION_CHAR	char(1)	one character id to be used as prefix, etc
#		

.....

1.3 PURCHASE_ORDERS

PO_ID	varchar(24)	formed from PO_NUMBER, BUYER_ID_MNEM, and PO_DATE to fully identify P.O.
#		
PO_NUMBER	varchar(10)	P.O. number issued by buyer
#		
BUYER_ID_MNEM	varchar(8)	mnemonic for buyer (e.g., SSC,FNL,IGC)
#		
PO_DATE	datetime	date of issue of P.O. to be entered in quotes "mm/dd/yy" no lead zeros required
#		
VENDOR_ID_MNEM	varchar(8)	supplier mnemonic
#		
NO_ITEMS	tinyint	number of separate line items on P.O.
#		
DELIVERY_DUE_DATE	datetime	date delivery is due at buyer; entered in standard format (see, e.g., PO_DATE)
#		

.....

1.4 PURCHASE_ORDER_ITEMS

PO_ID	varchar(24)	P.O. identification
#		
PO_ITEM_NUMBER	tinyint	number associated with line item on P.O.
#		
PO_ITEM_DESCRIPTION	varchar(50)	description of line on P.O. (e.g., blue ink pens, no. 2 pencils, etc)
#		
PO_ITEM_QUANTITY	int	how many of the line items we ordered (e.g., as in 3 gross of no. 2 pencils)
#		
PO_ITEM_UNITS	varchar(10)	single line-item delivery unit (e.g., gross,inches,etc)
#		

PO_COMMENTS	varchar(80)	any appropriate comments

.....

1.50 PRODUCT_SPEC_DOCS

PRODUCT_MNEM	varchar(20)	kind of product available from supplier (e.g., SC_INGOT,SSC_KAPTON,SSC_CABLE, RHIC_CB_CLOTH_EPOXY,SC_WIRE,etc)
#		
PROD_SPEC_CODE	varchar(50)	e.g., CDA-101000 for COPPER, KAPTON_TYPE_H_ FILM for SSC_KAPTON, HEXCEL_F185_STAGE_B for SSC_CB_CLOTH_EPOXY, CABLE_CLASS,etc
PROD_SPEC_DOC	varchar(50)	appropriate spec document; e.g., DUPONT_ BULL_GS-85-3 for KAPTON,SSC-MAG-M-4000 for SC_INGOT,SSC-MAG-M-4142 for CABLE_CLASS,etc
SPEC_DOC_REV_NO	varchar(2)	revision number (null for original)
DOC_DATE	datetime	date of document

.....

2.00 STRAND_CLASS_SPEC

WIRE_CLASS	varchar(10)	LAB_ID_MNEM-Wire Classification (e.g., SSC-4141, the 4141 refers to spec doc)
#		
ST_DOC_REV_NO	varchar(2)	strand doc revision no (null for original)
CU_PROD_MNEM	varchar(20)	kind of copper product directly specified, if any(CU_CAN,CU_HEX,etc, or just COPPER)
#		
CU_DOC_REV_NO	varchar(2)	copper doc revision no (null for original)
SC_PROD_MNEM	varchar(20)	kind of superconductor product specified, if any (e.g., SC_LOT, SC_INGOT)
#		
SC_DOC_REV_NO	varchar(2)	superconductor doc revision no (null for original)
ST_FILAMENT_SIZE #<microns<	float	diameter in microns (currently 6 for SSC)
ST_FILAMENT_SPACING #<microns<	float	minimum spacing between filaments (microns)
SURF_COND	varchar(50)	"no surface defects, inclusions, slivers, dirt, folds, laminations or NbTi filaments"

#

SHARP_BEND	varchar(50)	"no visible cracks at outer sharp bend"
#		
SHARP_BEND_ETCH	varchar(50)	"OK after nitric acid etch"
#		

.....

2.05 STRAND_TYPE_SPEC

STRAND_SPEC_ID	varchar(16)	full spec and id for strand, formed from [WIRE_CLASS]-[ST_USE_CODE][WIRE_CU_NONCU]-[WIRE_SUBTYPE] (e.g., SSC-4141-I15-B); this information will appear on every spool
#		
WIRE_CLASS	varchar(10)	LAB_ID_MNEM-Wire Classification
#		
ST_USE_MNEM	varchar(20)	mnemonic for intended use of strand (e.g., INNER_DIPOLE_STRAND, QUAD_STRAND, etc)
#		
ST_USE_CODE	char(1)	either dipole inner (I) or outer (O) or other, unspecified (U); obtained via look-up in ST_CB_USE_MNEMS table
#		
WIRE_CU_NONCU	tinyint	10x(Cu/NonCu Ratio) - since many of the strand characteristics are delineated by its value, the Cu/NonCu ratio can be used to specify the strand type; the ratio is specified to within 0.1 value
#		
WIRE_SUBTYPE	char(1)	any combination of possibilities, at present just of twist and anneal; A: Y-Y, B: Y-N, C: N-Y, and D: N-N; obtained via look-up in WIRE-SUBTYPES table
#		
ST_DOC_REV_NO	varchar(2)	different WIRE_CU_NONCU values appear in different DOC_REV_NO; this will keep track
ST_RMAX_295K	float	max resistance in micro ohm/cm at 295 K
#<micro-ohm/cm<		
ST_RMAX_10K	float	max resistance in micro ohm/cm at 9.5 K
#<micro-ohm/cm<		
SPRING_BACK	smallint	upper limit on springback angle (TM-4141-5)
#<degrees<		
ST_DIAM	float	spec value to within 0.0001 inch
#<inches<		
MAG_FIELD_LOWB	float	magnetic field in Tesla (usually 5.6)
#<Tesla<		
ST_ICMIN_LOWB	smallint	minimum critical current in amps for that low magnetic field
#<amps<		

MAG_FIELD_HIGHB #<Tesla<	float	magnetic field in Tesla (usually 7.0)
ST_ICMIN_HIGHB #<amps<	smallint	minimum critical current in amps for that high magnetic field
ST_TWIST_PITCH <twists/inch<	float	twists per inch to within 0.2 inches (null if no twist)
TWIST_ID #	varchar(4)	CW, CCW, NONE (clockwise, etc)
ANNEAL_STATE #	varchar(1)	type of wire: annealed=Y or not annealed=N
ANNEAL_RMAX_10K <micro-ohm/cm<	float	maximum resistance at 9.5 K after annealing of strand that comes unannealed (null if strand comes annealed)

.....

2.10 CABLE_TYPE_SPEC

CABLE_SPEC_ID #	varchar(16)	formed from CABLE_CLASS (e.g., SSC-4142)-CB_USE_CODE (e.g., I, O, Q) and.NO_STRANDS (e.g., SSC-4142-I23); the Cu/NonCu ratio is not specified, but can be traced via the STRAND SPEC ID
CABLE_CLASS #	varchar(10)	LAB MNEM and WIRE CLASS (e.g., SSC-4142)
CB_USE_CODE #	char(1)	one character stipulating use of cable (via look-up in ST_CB_USE_MNEMS table)
NO_STRANDS #	tinyint	number of strands in cable
CB_USE_MNEM #	varchar(20)	mnemonic for intended use of cable (e.g., INNER_DIPOLE_CABLE, QUAD_CABLE, etc)
STRAND_SPEC_ID #	varchar(16)	complete spec information on strand/wire
CB_DOC_REV_NO	varchar(2)	cable spec doc rev no
CB_LAY_PITCH #<inches<	float	specified in inches to within 0.2 inch
CB_LAY_DIR #	varchar(6)	"LEFT" for SSC
CB_WIDTH #<inches<	float	specified in inches to within 0.001 inch
CB_MID_THICK #<inches<	float	specified in inches to within 0.00025 inch

CB_KEYSTONE_ANGLE #<degrees<	float	specified in degrees to within 0.1 deg
ST_TWIST_PITCH #<twists/inch<	float	pitch of strand within formed cable, specified in twists/inch to within 0.5
MIN_WELDS_DIST #<feet<	smallint	distance in feet allowed between any two cold welds on cable (100 ft for SSC)
MIN_WELD_END #<feet<	smallint	distance in feet allowed for any cold weld to end of cable (250 ft for SSC)
WELD_TENS_STRNTH #<% of original<	tinyint	specified by some % of original wire, for SSC 50% of the tensile strength of wire
SURFACE_COND #	varchar(50)	SSC specifies smoothness, uniformity, etc.
CB_MAX_TWIST_ANGLE #<degrees<	tinyint	for SSC, 3 ft cable suspending 30 lb weight must curl by less than 90 degrees in lay direction (no twisting other way)
MAG_FIELD_LOWB #<Tesla<	float	magnetic field in Tesla (usually 5.6)
CB_ICMIN_LOWB #<amps<	smallint	minimum critical current in amps at low B
MAG_FIELD_HIGHB #<Tesla<	float	magnetic field in Tesla (usually 7.0)
CB_ICMIN_HIGHB #<amps<	smallint	minimum critical current in amps at high B
CB_RMAX_295K #<micro-ohms/cm<	float	resitance at 295 K in micro-ohms per cm
CB_RMAX_10K #<micro-ohms/cm<	float	resistance at 10 K in micro-ohms per cm

.....

2.20 ST_CB_USE_MNEMS

USE_MNEM #	varchar(20)	mnemonic for the strand or cable USE_CODE (QUAD_STRAND, INNER_DIPOLE_CABLE, etc)
USE_CODE #	char(1)	one character strand or cable use code (e.g, I for INNER_DIPOLE_STRAND, O for OUTER_DIPOLE_CABLE, etc)

.....

2.25 WIRE_SUBTYPES

TWIST_ID	varchar(4)	CW, CCW, NONE (clockwise, etc)
#		
ANNEAL_STATE	char(1)	type of wire: annealed=Y or not annealed=N
#		
WIRE_SUBTYPE	char(1)	calculated from any combination of above (or, later, other) possibilities; A: Y-Y, B: Y-N, C: N-Y, and D: N-N (for SSC, Y means CW)
#		

.....

2.30 CB_INSULATION_SPEC

CABLE_SPEC_ID	varchar(16)	completely specifies nature of cable (and is used as general spec code for tracking documents for wrapping the cable)
#		
INSULATION_CODE	varchar(12)	e.g., SK15-BF50, which specifies insulation used for any cable, e.g., single-wound (S) kapton (K) with 15% overlap, and butt-wound (B) prepreg fiberglass cloth (F) with 50% gap in the winding; formed from items below
#		
INSUL_WRAP_MNEM	varchar(20)	mnemonic for method used for wrapping the insulation (e.g., DUAL_WRAP, BUTT_WOUND, etc); also points to document
#		
INS_WRAP_DOC_REV_NO	varchar(2)	cable-insulation wrapping spec doc rev no
INSUL_WRAP_ID	char(1)	B for BUTT_WOUND, S for SINGLE_WOUND, I for INTERCALATED_WOUND, etc; obtained via look-up in INSULATION_MNEMS table
#		
INSUL_PERCENT_OVRLAP	tinyint	percent overlap of insulating tape
#		
CLOTH_WRAP_MNEM	varchar(20)	mnemonic used for method for wrapping fiberglass cloth (also points to document)
#		
CLOTH_WRAP_REV_NO	varchar(2)	cable cloth-wrapping spec doc rev no
CLOTH_WRAP_ID	char(1)	B for BUTT_WOUND, S for SINGLE_WOUND,etc; via look-up in INSULATION_MNEMS table
#		
CLOTH_PERCENT_GAP	tinyint	percent gap between fiberglass cloth tape
#		
INSUL_PROD_MNEM	varchar(20)	mnemonic for insulator; e.g., SSC_KAPTON (points to document and vendor products table for material)
#		
INSUL_TYPE_REV_NO	varchar(2)	cable insulation-material spec doc rev no

CB_INSUL_ID	char(1)	K for SSC_KAPTON tape, etc; obtained via look-up in INSULATION_MNEMS table
#		
CB_CLOTH_MNEM	varchar(20)	mnemonic for type cloth; e.g., B_STGE_EPOX_ FIBRGLSS; (points to document and vendor products table for material)
#		
CLOTH_TYPE_REV_NO	varchar(2)	cable cloth-material spec doc rev no
CB_CLOTH_ID	char(1)	F for B-stage prepreg fiberglass tape, etc; via look-up in INSULATION_MNEMS table
#		
CB_CLOTH_FINISH	varchar(20)	mnemonic for treatment with e.g, silane enhancement Z6040, or Hexcel F69, etc. (points to document and product table); e.g., HEXCEL_F69, SILANE_Z6040, etc
FINISH_TYPE_REV_NO	varchar(2)	finish on cable spec doc rev no
CB_EPOXY_MNEM	varchar(20)	mnemonic for epoxy used in cloth (points to document and products table)
#		
CB_EPOXY_REV_NO	varchar(2)	cable eppoxy spec doc rev no

.....

2.35 INSULATION_MNEMS

INSULATION_MNEM	varchar(20)	method or material used for wrapping cable; e.g.,SSC_KAPTON, DUAL_WRAP, BUTT_WOUND, etc
#		
INSUL_MNEM_ID	char(1)	B for BUTT_WOUND, S for SINGLE_WOUND, I for INTERCALATED_WOUND, K for SSC_KAPTON, F for B_STGE_EPOX_FIBRGLSS, etc
#		

.....

3.00 SC_INGOTS

VENDOR_ID_MNEM	varchar(8)	manufacturer's or supplier's mnemonic
#		
INGOT_HEAT_NUMBER	varchar(20)	ingot or "melt" heat-treatment number for NbTi rod, or certification number if for superconductor cladding material
#		
INGOT_SPEC_DOC	varchar(20)	appropriate spec document
SPEC_DOC_REV_NO	varchar(2)	document revision number

3.05 INGOT_CHEM_AN
==========

VENDOR_ID_MNEM	varchar(8)	manufacturer's or supplier's mnemonic
#		
INGOT_HEAT_NUMBER	varchar(20)	ingot or "melt" heat-treatment number (or cladding certification number)
#		
MAX_PPM_AL_TOP	smallint	maximum parts per million of aluminum by weight found at top of ingot
<ppm by weight<		
MAX_PPM_C_TOP	smallint	
<ppm by weight<		
MAX_PPM_CR_TOP	smallint	
<ppm by weight<		
MAX_PPM_CU_TOP	smallint	
<ppm by weight<		
MAX_PPM_FE_TOP	smallint	
<ppm by weight<		
MAX_PPM_H_TOP	smallint	
<ppm by weight<		
MAX_PPM_N_TOP	smallint	
<ppm by weight<		
MAX_PPM_NI_TOP	smallint	
<ppm by weight<		
MAX_PPM_O_TOP	smallint	
<ppm by weight<		
MAX_PPM_SI_TOP	smallint	
<ppm by weight<		
MAX_PPM_TA_TOP	smallint	
<ppm by weight<		
MAX_PPM_AL_BOT	smallint	maximum parts per million of aluminum by weight found at bottom of ingot
<ppm by weight<		
MAX_PPM_C_BOT	smallint	
<ppm by weight<		
MAX_PPM_CR_BOT	smallint	
<ppm by weight<		
MAX_PPM_CU_BOT	smallint	
<ppm by weight<		
MAX_PPM_FE_BOT	smallint	
<ppm by weight<		
MAX_PPM_H_BOT	smallint	

<ppm by weight<

MAX_PPM_N_BOT smallint
<ppm by weight<

MAX_PPM_NI_BOT smallint
<ppm by weight<

MAX_PPM_O_BOT smallint
<ppm by weight<

MAX_PPM_SI_BOT smallint
<ppm by weight<

MAX_PPM_TA_BOT smallint
<ppm by weight<

PERCENT_TI_TOP float percent titanium by weight at top of ingot
(null, if reported as ballance from % Nb)
<% by weight<

PERCENT_NB_TOP float percent niobium by weight at top of ingot
(null, if reported as ballance from %Ti)
<% by weight<

PERCENT_TI_BOT float
<% by weight<

PERCENT_NB_BOT float
<% by weight<

QUAL_ASSURE_REP varchar(20) name of rep from qual. assurance dept.
#

QUAL_ASSURE_DATE datetime date of quality assurance for lot
#

.....

3.10 SC_LOTS

VENDOR_ID_MNEM varchar(8) manufacturer's or supplier's mnemonic
#

INGOT_HEAT_NUMBER varchar(20) ingot or "melt" heat-treatment number
#

SC_LOT_ID varchar(12) superconductor lot or "macs" no.
#

PO_ID varchar(24) P.O. identification
#

BUYER_ID_MNEM varchar(8) mnemonic of buyer (e.g., SSC,FNL,IGC)
#

QUAL_ASSURE_REP varchar(20) name of rep from qual. assurance dept.
#

QUAL_ASSURE_DATE datetime date of quality assurance for lot
#

PO_ITEM_NUMBER tinyint number associated with line item on P.O.
#

SC_LOT_SPEC_DOC	varchar(20)	appropriate spec document
SPEC_DOC_REV_NO	varchar(2)	document revision number

.....

3.15 SC_LOT_CHEM_AN
==========

VENDOR_ID_MNEM #	varchar(8)	manufacturer's or supplier's mnemonic
INGOT_HEAT_NUMBER #	varchar(20)	ingot or "melt" heat-treatment number
SC_LOT_ID #	varchar(12)	lot or macs no.
PPM_O <ppm by weight<	smallint	
PPM_H <ppm by weight<	smallint	
PPM_C <ppm by weight<	smallint	
PPM_N <ppm by weight<	smallint	
QUAL_ASSURE_DATE #	datetime	date of quality assurance for lot
QUAL_ASSURE_REP #	varchar(20)	name of rep from qual. assurance dept.

.....

3.20 COPPER_MATERIAL
==========

VENDOR_ID_MNEM #	varchar(8)	manufacturer's or supplier's mnemonic
CU_HEAT_NUMBER #	varchar(20)	OFHC copper heat identification number
CU_LOT_ID #	varchar(12)	manufacturing number for copper lot
PO_ID #	varchar(24)	P.O. identification
PO_ITEM_NUMBER #	tinyint	number associated with line item on P.O.
PRODUCT_MNEM #	varchar(20)	kind of copper product (CU_CAN,CU_HEX,etc)
CU_SPEC_DOC	varchar(20)	appropriate spec document
SPEC_DOC_REV_NO	varchar(2)	document revision number

QUAL_ASSURE_REP	varchar(20)	name of rep from qual. assurance dept.
#		
QUAL_ASSURE_DATE	datetime	date of quality assurance for lot
#		

.....

3.25 COMPOSITE_BILLET

BL_ID_VENDOR	char(13)	billet code number formed from vendor info (this can be specified by: VENDOR ID CHAR, hyphen, VENDOR JOB NO, hyphen, 4-digit MFG BILLET NO, hyphen, billet part)
#		
VENDOR_ID_CHAR	char(1)	manufacturer's or supplier's id character
#		
VENDOR_JOB_NO	smallint	manufacturer's job number
#		
MFG_BL_NO	smallint	billet number on certificate of conformance
#		
BILLET_PART	char(1)	specifies which section (presently either A or B; if just one section, use A); set to R for the case when the billet is used for forming composite sc rod for restacking
#		
BL_EXTR_DATE	datetime	billet extrusion date, not needed for now
BL_SER_ID	varchar(8)	billet serial number assigned by buyer, not needed for SSC at this time
PO_ID	varchar(24)	P.O. identification
#		
PO_ITEM_NUMBER	tinyint	number for this line item (billet) on P.O.
#		
BL_CLASS_MNEM	varchar(20)	mnemonic for class of billet (usually for some particular kind of strand), e.g., BL_SSC-4141; points to spec document (if one exists)
#		
SPEC_DOC_REV_NO	varchar(2)	document revision number
VENDOR_PLANT_ID	varchar(20)	plant where extrusion took place
#		
QUAL_ASSURE_DATE	datetime	date on billet certificate of conformance
#		
QUAL_ASSURE_REP	varchar(20)	signer of certification for assurance of quality
#		
BILLET_USE_MNEM	varchar(20)	use for billet, eg., FOR_RESTACKING, FOR_DRAWING_TO_WIRE, etc.
#		

BILLET_COMMENT	varchar(150)	any appropriate remarks

.....

3.30 COMPOSITE_BL_COMPONENTS

"There will be several rows entered for any particular billet. The number and kind will depend on the nature of the manufacturing process. The simplest would be for that of unclad superconductor rod (SC_ROD) that is stacked in, say, a hexagonal copper matrix (CU_HEX), with the front, rear, and perimeter of the stack covered (welded) with copper material (CU_NOSE, CU_TAIL, and CU_CAN, in respective order). In this case, there will be at least one row for each of the above fields, specified in, say, the above order. Each row would list, along with BUYER_ID, BL_ID_VENDOR, BL_PROD_MNEM, and the SPEC_DOC_REV_NO, the full BL_COMPONENT_NO; the latter would be different for material with different batch lots or heat numbers that contributed to the billet stack. As example, I have below a stack made from two superconductor batches, two copper lots for the matrix, a different copper lot for the periphery, another different copper lot for the nose and tail. BUYER_ID and BL_ID_VENDOR will be the same for all entries for any single billet, while BL_PROD_MNEM and SPEC_DOC_REV_NO should repeat when any component comes from more than one source. Thus we get:

BL_PROD_MNEM	BL_COMPONENT_NO
SC_ROD	IGC-9I87J-1328L
SC_ROD	IGC-7I7J-1208L
CU_HEX	WAH-9OL870D-769
CU_HEX	WAH-1OL8730D-24
CU_CAN	USD-973P-375P
CU_NOSE	JFK-1374-458F
CU_TAIL	JFK-1374-458F

(For other entries - see above.)

Note that for SC_ROD we use VENDOR_HEAT#-MACS# and for copper VENDOR_HEAT#-LOT#. We do not record what fraction of any material makes up the billet, but just identify each pertinent heat and lot number (or the conformance documents)."

BUYER_ID #	varchar(8)	id of buyer (e.g., SSC,FNL,IGC)
BL_ID_VENDOR #	char(13)	billet code number
BL_PROD_MNEM #	varchar(20)	set of mnemonics (eg., one of: SC_ROD, CLAD_ROD,COMPOS_ROD; then: CU_CAN,CU_TAIL, & CU_NOSE; then one of: CU_HEX, CU_TUBE, or OTHER) that specify all the contents, and repeat until all contents are tagged to a source (also point to docs)
SPEC_DOC_REV_NO	varchar(2)	document revision number
BL_COMPONENT_NO	varchar(40)	the heat and lot numbers that tag mnemonics to their material tables; for simple billet see TABLE_COMMENTS; for cladding use VENDOR _CLAD_CERTIF (from heat no. location in SC_ INGOTS); for composite billet point to right BL_ID_VENDOR and to any other copper

#

.....

3.35 BILLET_TEST_INFO

BL_ID_VENDOR	char(13)	billet code number from vendor info
#		
NUMBER_FILAMENTS	int	approximate number of elements in billet
BL_CU_NONCU_BEG	float	Cu/NonCu ratio for first strand of billet
BL_CU_NONCU_END	float	Cu/NonCu ratio for last strand of billet
ST_FIL_SIZE_BEG	float	diameter in microns for first strand
<microns<		
ST_FIL_SIZE_END	float	diameter in microns for last strand
<microns<		
ST_FIL_SPACING_BEG	float	minimum spacing between filaments (microns) for first strand on billet
<microns<		
ST_FIL_SPACING_END	float	minimum spacing between filaments (microns) for last strand on billet
<microns<		
CERTIF_COMPLIANCE	varchar(12)	certificate number for test quality
#		
CERTIF_DATE	datetime	date for tests
#		
TEST_SPRVSR	varchar(20)	supervisor of tests
#		
TEST_DOCUMENT	varchar(20)	document in which test is specified

.....

3.4 STRAND_PRODUCTION

BL_ID_VENDOR	char(13)	billet code number from vendor info
#		
SPOOL_NUMBER	smallint	spool number for any billet from vendor
#		
ST_PROD_ID	char(17)	this is formed from BL_ID_VENDOR and the SPOOL_NUMBER supplied by vendor
#		
STRAND_SPEC_ID	varchar(16)	full spec and id for strand/wire
#		

SUPPLIER_ID #	varchar(8)	strand supplier id (usually same as billet)
BL_SER_ID #	varchar(8)	billet serial number (assigned if needed)
SPECIAL_DRAW_SPEC	varchar(50)	any special specs for drawing the wire (for use in R&D, if needed)
ST_LENGTH_FROM_VENDOR #<feet<	int	length of wire on spool (in feet)
ST_WT_FROM_VENDOR #<pounds<	float	net weight of spool (in lbs)
ST_DRAW_COMPL_DATE #	datetime	if needed (date on qual assurance certif for billet should be good enough)
ST_PO_ID #	varchar(24)	P.O. ID of original buyer (usually same as billet po id)
ST_USE_PLAN	varchar(50)	stipulates planned use: CABLE, SET ASIDE, CORRECTION COIL, R&D, etc
ORIG_LOCATION_CODE #	varchar(8)	ID of lab or plant where spool is first shipped and stored
ST_ANNEAL_DATE	datetime	if strand is not originally annealed (i.e., subtype B or D), otherwise null
ST_ANNEAL_PLANT	varchar(20)	plant mnemon, or null (if not required)
ST_TWIST_DATE	datetime	if strand is not originally twisted (i.e., subtype C or D), otherwise null
ST_TWIST_PLANT	varchar(20)	plant mnemon, or null (if not required)
ST_COATING_MNEM	varchar(20)	mnemonic for coating material, if any used; this also points to document table
COAT_DOC_REV_NO	varchar(2)	
ST_COATING_DATE	datetime	
ST_COATING_PLANT	varchar(20)	

.....

3.5 STRAND_RECEIVING

"Only the wire that we deal with directly will appear in this table. After the initial entry here, the spools will be tracked through the SPOOL_INVENTORY_USE table."

ST_PROD_ID #	char(17)	this is formed from vendor info (3.4)
BUYER_ID_MNEM #	varchar(8)	cabler or other user (mnemonic)

VENDOR_ID_CODE	char(1)	character representing vendor
#		
STRAND_SPEC_ID	varchar(16)	full spec and id for strand/wire
#		
ST_USE_FINAL	varchar(20)	stipulates final use: SET ASIDE, SHORT SAMPLE (test for cable), CORRECTION COIL, DIPOLE-INNER, DIPOLE-OUTER, etc
ST_USE_CODE	char(1)	character representing use (e.g., I,O,etc)
ST_SERIAL_NO	int	serial number assigned by us (see below)
#		
ST_SEG	varchar(3)	segment number cut off spool (with code for just general location: hub/inside=END and loose/outer=OUT) original full spool gets the code=00 entered automatically, after cutting ST_SEG id changes as for cable (see comments for CB_PROD_INVENTORY table)
#		
ST_SPOOL_ID	varchar(21)	this is formed from vendor info ST PROD ID, hyphen, ST_SEG id, unique for any piece
#		
BUYER_SPOOL_ID	varchar(17)	equivalent to our CABLE ID (we assign a code: BUYER_ID_MNEM-ST_USE_CODE-VENDOR_ID_CODE-ST_SERIAL_NO-ST_SEG for strand that we test or inventory for future)
#		
ST_DATE_RECEIVED	datetime	
#		
ST_RECEIVED_OPERATOR	varchar(20)	person who signed for item
#		
ST_RECEIVED_LOCATION	varchar(8)	location id of place strand is received
#		
CONTACT_PERSON	varchar(20)	person responsible for the spool of strand
ST_LENGTH_RECEIVED	int	in feet
#		
ST_RECEIVED_COMMENTS	varchar(50)	whatever is appropriate

.....

4.0 STRAND_ELEC_TEST

"Tests are performed at parameter values close to those desired by the user, and extrapolated to standard (reported) settings that appear in this table. For SSC low field is 5.6 T and high is 7.0 T. Tests are specified in SSC-MAG-M-4141."

ST_SPOOL_ID	varchar(21)	this is formed from vendor info (ST_PROD_ID

		and ST_SEG)
#		
ST_BUYER_SPOOL_ID	varchar(17)	equivalent to our CABLE ID (when null, then test piece or for routine use in cabling); obtained via look-up in STRAND_RECEIVING
#		
ST_ELEC_RUN	int	run number for electrical test
#		
TEST_LOC_MNEM	varchar(8)	either manufacturer's or our own mnemonic
#		
TEST_DATE	datetime	
#		
OPERATOR_ID	varchar(20)	manager of quality control, or tech doing the measurements
#		
MAG_FIELD_LOWB <Tesla<	float	magnetic field in Tesla (usually 5.6)
TEMP_STRAND_LOWB <Kelvin<	float	temperature of sample (usually 4.22 K)
IC_STRAND_LOWB <amps<	float	critical current in amps for that field
JC_STRAND_LOWB <amps/sq-cm<	float	critical current density in amps per sq mm for that field (calculated)
IQ_STRAND_LOWB <amps<	float	quench current in amps for that field
QUAL_INDEX_N_LOWB	smallint	from V-I curve at that field
MAG_FIELD_HIGHB <Tesla<	float	magnetic field in Tesla (usually 7.0)
TEMP_STRAND_HIGHB <Kelvin<	float	temperature of sample (usually 4.22 K)
IC_STRAND_HIGHB <amps<	float	critical current in amps for that field
JC_STRAND_HIGHB <amps/sq-cm<	float	critical current density in amps per sq mm for that field (calculated)
IQ_STRAND_HIGHB <amps<	float	quench current in amps for that field
QUAL_INDEX_N_HIGHB	smallint	from V-I curve at that field
ST_RES_295K <micro-ohm/cm<	float	strand resistance in micro ohm/cm at 295 K
ST_RES_10K <micro-ohm/cm<	float	strand resistance in micro ohm/cm at 9.5 K
ST_ANNEAL_RES_10K <micro-ohm/cm<	float	strand resistance at 9.5 K after annealing

ST_ELEC_CU_NONCU	float	Cu/NonCu ratio calculated by operator from R10 & R295 via spec in ISA Tech Note 385
ST_RRR	float	R at 295K divided by R at 9.5K (calculated)

4.1 STRAND_CHEM_MECH_TESTS

"For the SSC, all tests are specified in appendixes of document SSC-MAG-M-4141."

ST_SPOOL_ID #	varchar(21)	this is formed from vendor info (ST_PROD_ID and ST_SEG)
ST_BUYER_SPOOL_ID #	varchar(17)	equivalent to our CABLE ID (when null, then test piece or for routine use in cabling); obtained via look-up in STRAND_RECEIVING
TEST_LOC_MNEM #	varchar(8)	either manufacturer's or our own mnemonic
TEST_DATE #	datetime	
OPERATOR_ID #	varchar(20)	manager or tech doing test
SPRING_BACK <degrees<	smallint	springback angle in degrees
ST_DIAM <inches<	float	spec value to within 0.0001 inch
ST_TWIST_PITCH <twists/inch<	float	twists per inch
TWIST_ID	varchar(4)	CW, CCW, NONE (clockwise, etc)
ULT_TENSILE_STRNGTH <kpsi<	float	in thousands psi
PERCENT_ELONG <%<	float	
CU_NONCU	float	by weight (using chemical means)
SURF_COND	varchar(50)	describe relative to spec
SHARP_BEND	varchar(50)	describe relative to spec (three times)
SHARP_BEND_ETCH	varchar(50)	describe relative to spec (after each of three nitric acid etchings)

4.5 STRAND_INSULATION

"Here we can put any information pertaining to insulation or similar treatment of special strand to be used for correction coils, etc."

ST_SPOOL_ID	varchar(21)	this is formed from vendor info (ST_PROD_ID

and ST_SEG)

#

ST_INSUL_MNEM	varchar(20)	mnemonic for insulating strand, if used; this also points to document table
#		
ST_INSUL_REV_NO	varchar(2)	
ST_LNGTH_INSUL	int	
#<feet<		
DATE_ST_INSUL	datetime	
#		

.....

5.00 CABLE_PRODUCTION

"This table provides information about spools of bare cable that have been manufactured and are ready for insulation. One row of entry per unique spool."

CABLE_PROD_ID	char(13)	formed from CB BUYER MNEM-CB USE CODE-CB VENDOR ID CHAR-CB SERIAL NUMBER (e.g., SSC-0-F-00055)
#		
CB_BUYER_MNEM	varchar(8)	mnemonic for buyer (SSC,BNL,FNL,DSY...)
#		
CB_USE_CODE	char(1)	one character stipulating anticipated use of cable and the number of strands in that cable (see ST_CB_USE_MNEMS table for mnems)
#		
CB_VENDOR_ID_CHAR	char(1)	one character for id of manufacturer (see VENDORS table for codes)
#		
CB_SERIAL_NUMBER	int	serial number assigned to any cable that is processed through our system; as usual, leading zeroes to be used when combined in full ID
#		
CB_PO_ID	varchar(24)	purchase order id for cable
#		
CABLE_SPEC_ID	varchar(16)	
#		
CB_DOC_REV_NO	varchar(2)	
CB_PRODUCTION_DATE	datetime	
#		
CB_QUAL_ASSURE_REP	varchar(20)	
#		
CB_PRODUCTION_LENGTH	int	usually close to 12,000 feet

#<feet<

CB_WIRE	varchar(50)	compact summary of info on billet and wire manufacturer - for reports and not look-up (e.g., HITACHI CU, WAH CHANG SC, IGC BL)
COMMENTS	varchar(150)	pertinent remarks

.....

5.05 CABLE_STRAND_MAPS

"As cable production begins, the strand map starts with a row of entries for each spool positioned on any spool-holder of the cabling machine. Strand welds located on these spools, that join together different segments, are recorded along with appropriate original spool numbers (and segment numbers), and are entered as new rows. When cabling is completed, a final entry is made for each remaining spool of strand. (If any strand should break during cabling, cable production is halted, and a new set of spools is mounted.)"

CABLE_PROD_ID	char(13)	cable id (before insulation)
#		
SPOOL_HOLDER	tinyint	spool-holder number on cabling machine
#		
ST_PROD_ID	char(17)	produced strand spool id for above holder
#		
ST_SEG_VENDOR_ID	varchar(3)	segment number cut off spool (provided by vendor, primarily for his own inventory)
#		
ST_LNGTH_SPOOL_BEG	int	linear footage reading from strand spool at beginning of this cable segment
#<feet<		
CB_LNGTH_SPOOL_BEG	int	linear footage on cable spool at beginning of this strand segment
#<feet<		
COLD_WELD_DATE	datetime	date of cold weld (beginning date for start of cable, end date for end of production)
#		
COLD_WELD_SUPERVISOR	varchar(20)	quality assurance supervisor
#		
TENSILE_STR_CERT_NO	varchar(10)	certification number for test of tensile strength of cold weld (null for first and final entry)
#		
ST_LNGTH_SPOOL_END	int	linear footage reading from strand spool at end of this cable segment
#<feet<		
CB_LNGTH_SPOOL_END	int	linear footage on cable spool at end of this strand segment
#<feet<		

.....

5.10 CB_PROD_INVENTORY

"Here we keep track of any unwrapped cable in storage and test samples sent to us from the manufacturer. We first enter the amount received or produced (as given in CABLE_PRODUCTION table). As the cable is cut and used up, we track the original spool and any progeny spools. This is done by tracking a sequential cut number (CUT_NO) and the source segment (SOURCE_NO) for that cut. First removed segment is given CB_SEG of 01. If a useful length of the original spool (00) is returned to storage, a new row is entered with this CABLE_ID and its remaining length, so that it can be recycled. If any part of the cut-off segment (01) is later returned to storage, it will also be entered as a new row with that very same CB_SEG id. If the original spool or that segment are cut again, the next progeny-segment will be labeled X2, where X is either 0 (original spool) or 1 (the first segment) etc. The cutting process can continue until a total of 35 cuts have been made (0-Z). There may be different row entries for same CABLE ID, that will differ, however, in dates of entry and feet of cable (daisy chained). The last entry with same id will give amount of cable remaining on that spool."

CABLE_ID	varchar(17)	formed from next two, separated by hyphen
#		
CABLE_PROD_ID	char(13)	formed from CB BUYER MNEM-CB USE CODE-CB VENDOR ID CHAR-CB SERIAL NUMBER (e.g., SSC-O-F-00545)
#		
CB_SEG	varchar(3)	END or BEG for test piece, automatic 00 for start of spool; as the cable is used up, this original segment id never gets changed!
#		
CB_LENGTH	int	length of cable in feet
#<feet<		
CABLE_STORAGE	varchar(8)	mnemonic location where cable is stored
#		
CABLE_STORE_DATE	datetime	
#		
CB_RECEIVE_SPRVSR	varchar(20)	
#		
CB_REMOVE_DATE	datetime	
CB_USE_SPRVSR	varchar(20)	
CB_INVENTORY_MNEM	varchar(20)	RECEIVED, TEST_SAMPLE, SENT_TO_WRAP, etc
#		
INVENTORY_CODE	char(1)	W if taken to wrapping line, X if rejected, D if used in R&D, etc. R if just stored and ready for use elsewhere; via look-up
#		
CB_REMOVE_COMMENT	varchar(150)	any appropriate comments
CB_COMMENT_PERSON	varchar(20)	person responsible for entering COMMENT
CB_DESTINATION	varchar(8)	lab mnemonic where cable segment is shipped

CB_RESPONSIBLE	varchar(20)	individual shipped to

.....

5.15 CABLE_COATING

"After a cable is manufactured, it may have to be tinned or dipped, etc., before it is wrapped. This phase can be tracked in this table."

CABLE_ID	char(17)	
#		
CB_COAT_MNEM	varchar(20)	
#		
CB_COAT_REV_NO	varchar(20)	

.....

5.20 CABLE_INSULATION

"After manufacture, cable is insulated. The CABLE INSULATION table has a row for each cable's insulation and wrapping step. When lumps or pinholes are sensed to be out of range, a new row is entered. If a cable must be cut during wrapping, the id of the part still on the original spool is not affected, but the cut-off piece gets a new CB_SEG (and thus new CABLE_ID as well). The id is formed as described in the CB_PROD_INVENTORY table"

CB_WRAPLINE_ID	varchar(8)	mnemonic for wrapping plant and machine
#		
INSUL_PO_ID	varchar(24)	purchase order id for insulation of cable
#		
CABLE_ID	char(17)	
#		
INSULATION_CODE	varchar(12)	e.g., SK15-BF50, which specifies insulation (for checking with info below, which is entered online)
#		
INSUL_WRAP_MNEM	varchar(20)	mnemonic for method used for wrapping the insulation (e.g., DUAL_WRAP, BUTT_WOUND, etc); also points to document
#		
INSUL_WRAP_ID	char(1)	B for BUTT_WOUND, S for SINGLE_WOUND, I for INTERCALATED_WOUND, etc; obtained via look-up in INSULATION_MNEMS table
#		
INSUL_PERCENT_OVRLAP	tinyint	percent overlap of insulating tape
#<%<		

CLOTH_WRAP_MNEM	varchar(20)	mnemonic used for method for wrapping fiberglass cloth (also points to document)
#		
CLOTH_WRAP_ID	char(1)	B for BUTT_WOUND, S for SINGLE_WOUND,etc; via look-up in INSULATION_MNEMS table
#		
CLOTH_PERCENT_GAP	tinyint	percent gap between fiberglass cloth tape
#<%<		
INSUL_PROD_MNEM	varchar(20)	mnemonic for insulator; e.g., SSC_KAPTON (points to document and vendor products table for material)
#		
CB_INSUL_ID	char(1)	K for SSC_KAPTON tape, etc; obtained via look-up in INSULATION_MNEMS table
#		
CB_CLOTH_MNEM	varchar(20)	mnemonic for type cloth; e.g., B_STGE_EPOX_ FIBRGLSS; (points to document and vendor products table for material)
#		
CB_CLOTH_ID	char(1)	F for B-stage prepreg fiberglass tape, etc; via look-up in INSULATION_MNEMS table
#		
CB_CLOTH_FINISH	varchar(20)	mnemonic for treatment with e.g, silane enhancement Z6040, or Hexcel F69, etc. (points to document and product table); e.g., HEXCEL_F69, SILANE_Z6040, etc
CB_EPOXY_MNEM	varchar(20)	mnemonic for epoxy used in cloth (points to document and products table)
#		
CB_LNGHT_BEG	int	
#<feet<		
INSUL_LNGTH_BEG	int	
#<feet<		
INSUL_MAT_CERTIF	varchar(20)	mfg mnemonic and mill roll number
#		
CLOTH_LNGTH_BEG	int	
#<feet<		
EPOX_CLOTH_CERTIF	varchar(20)	mfg mnemonic and batch lot number
#		
START_TIME	datetime	starting date and time in sybase format
#		
STOP_CODE	varchar(20)	LUMP IN INPUT, KAPT RUNOUT, CLOTH RUNOUT, PINHOLE, LUMP IN OUTPUT, CB START, CB END
#		
STOP_TIME	datetime	starting date and time in sybase format
#		
CB_LNGTH_END	int	
#<feet<		

INSUL_LNGTH_END #<feet<	int	
CLOTH_LNGTH_END #<feet<	int	
ACTION_MNEM #	varchar(50)	KAPTON SPLICE, CABLE CUT, COLD STORE, etc
QUAL_ASSUR_SUPVSR #	varchar(20)	

.....

5.45 CB_ACTION_CODES

INVENTORY_CODE #	char(1)	R=received, T=test sample, C=for coil mfg, S=shipped elsewhere, X=reject, W=wrapping line, etc.
CB_INVENTORY_ACTION #	varchar(20)	RECEIVED, TEST SAMPLE, SENT TO WRAP, SENT TO COIL, etc

.....

5.5 SPOOL_INVENTORY_USE

"One row for each piece of special strand or insulated cable. The entries here are initiated when some authorized individual requests special strand for coil production (or similar purpose) to be set aside, or when insulated cable is produced and is ready for storage, also when a partially used spool is returned for inventory.) We use the same tracking scheme as in CB_PROD_INVENTORY table"

CABLE_ID #	varchar(17)	full strand or wrapped cable id
CB_LENGTH #<feet<	int	starting length in feet
MATERIAL_SPEC_ID #	varchar(16)	full spec for strand or cable (this can be obtained via lookup in STRAND_RECEIVING or CABLE_PPRODUCTION tables)
INSULATION_CODE #	varchar(12)	null if bare material or strand
CABLE_STORAGE #	varchar(8)	mnemonic location where cable is stored
CB_INVENTORY_DATE #	datetime	date of start of action (or date that cable entered into cold store)
CB_STORE_SUPVSR	varchar(20)	individual responsible for proper storage

#		
CB_INVENTORY_MNEM	varchar(20)	RECEIVED, TEST_SAMPLE, SENT_TO_WRAP, etc
#		
INVENTORY_CODE	char(1)	R=received and ready for use,T=test sample, C=sent for coil mfg, S=shipped elsewhere, X=rejected, etc. (via look-up)
#		
CB_REMOVE_DATE	datetime	
CB_USE_SPRVSR	varchar(20)	
CB_REMOVE_COMMENT	varchar(150)	any appropriate remarks
CB_COMMENT_PERSON	varchar(20)	person responsible for entering COMMENT
CB_DESTINATION	varchar(8)	lab mnemonic where segment is being shipped
CB_RESPONSIBLE	varchar(20)	individual shipped to
CB_SEGMENT_OWNER	varchar(20)	name of person responsible for new segment
CB_INVENTORY_BY	varchar(20)	name of person who made this entry
CB_INVENTORY_NOTE	varchar(250)	general catch-all for untablifiable remarks

.....

6.0 CB_ELEC_SUMMARY

"For SSC, this table contains results calculated for standard B-field values, usually 5.6 and 7.0 Tesla at 4.22 K, and the resistance-related measurements at B=0. The appropriate test specifications can be found in SSC_MAG_M_4142."

CABLE_ID	varchar(17)	standard cable ID
#		
CB_SAMPLE_NOTE	varchar(20)	general location on initial cable
#		
CB_ELEC_RUN	int	run # for cable electrical test
#		
CB_ELEC_DATE	datetime	date of cable electrical test
#		
MAG_FIELD_LOWB <Tesla<	float	magnetic field in Tesla (usually 5.6)
CB_IC_LOWB <amps<	int	calc. cable critical current (amps) usually at 5.6T and 4.22K
CB_JC_LOWB <amps/sq-mm<	int	calc. cable critical current density (amps/mm2) usually at 5.6T and 4.22K
MAG_FIELD_HIGHB	float	magnetic field in Tesla (usually 7.0)

<Tesla<

CB_IC_HIGHB	int	calc. cable critical current (amps) usually at 7.0T and 4.22K
<amps<		
CB_JC_HIGHB	int	calc. cable critical current density (amps/mm2) usually at 7.0T and 4.22K
<amps/sq-mm<		
CB_R295	float	cable resistance (uohms/cm) at 295K, B=0
<micro-ohms/cm<		
CB_R10	float	cable resistance (uohms/cm) at 10K, B=0
<micro-ohms/cm<		
CB_RRR	int	calculated cable residual resistance ratio
CB_ELEC_CU_NONCU	float	cable copper to superconductor ratio from electrical meas. at B=0
CB_ELEC_NOTE	varchar(50)	comments about cable electrical test
CB_ELEC_ENTRY	varchar(20)	who entered these data, and where (lab id)
CB_ELEC_PROC_DOC	varchar(20)	document that prescribes procedures and their analysis
DOC_REV_NO	varchar(2)	document revision number

.....

6.1 CB_ELEC_MEAS

"Measurements at misc. field values (usually near B=5.0, 5.6 and 6.0 Tesla), and some temperature near 4.2 K. The appropriate test specifications can be found in SSC_MAG_M_4142."

CABLE_ID	varchar(17)	standard cable ID
#		
CB_SAMPLE_NOTE	varchar(20)	general location on initial cable
#		
CB_ELEC_RUN	int	run # for cable electrical test
#		
CB_ELEC_OPERATOR	varchar(20)	name and location of test operator
#		
CB_ELEC_DATE	datetime	date of cable electrical test
#		
CB_ELEC_B_PEAK	float	peak field (Tesla) for cable elect. test = applied field + self field
<Tesla<		
CB_ELEC_TEMP	float	temperature (K) for cable elec. meas.
<Kelvin<		
CB_ELEC_IT	int	critical current (amps) at field & temp
<amps<		
CB_ELEC_IQ	int	cable quench current (amps) at field & temp

<amps<

CB_ELEC_N	int	calculated cable elec. meas. quality index for above conditions (from V-I curve)

.....

6.2 CB_MAGNETIZATION

"Cable magnetization measurements may be performed at different B-fields (all at 4.2 K). We allow one row per ramp-rate and field value."

CABLE_ID	varchar(17)	standard cable ID
#		
CB_SAMPLE_NOTE	varchar(20)	general location on initial cable
#		
CB_MAGNTZ_RUN	int	run # for cable magnetization tests
#		
CB_MAGNTZ_DATE	datetime	date of cable magnetization tests
#		
B_FIELD_RAMP	tinyint	dB/dt in mTesla/sec (usually near 21)
#<mTesla/sec<		
B_FIELD	float	at which measurement is recorded
#<Tesla<		
WIDTH_MAGNTZ_LOOP	float	width of magnetization loop (mT) usually at that field (usually B=0.3T)
<mTesla<		
EDDY_MAGNTZ_WIDTH	float	width of eddy current magnetization (mT) usually at B=0.3T
<mTesla<		
CB_MAGNTZ_NOTE	varchar(50)	comment for magnetization meas.
CB_MAGNZ_BY	varchar(20)	who did magnetization meas. and where
#		

.....

6.3 CB_MECHANICAL_MEAS

"For SSC, this table contains results of tests stipulated in specifications in appendixes of SSC_MAG_M_4142. "

CABLE_ID	varchar(17)	standard cable ID
#		
CB_SAMPLE_NOTE	varchar(20)	general location on initial cable
#		
INLINE_ID	char(1)	1 if inline tests, 0 if offline; for ID=0 there will clearly be fewer entries
#		
CB_MECH_BY	varchar(20)	who did mechanical meas. and where (lab

		mnemonic and name of individual or machine)
#		
CB_MEAS_DATE	datetime	
#		
MIN_PRESSURE <kpsi<	float	in kpsi, minimum hydraulic pressure sensed
MAX_PRESSURE <kpsi<	float	in kpsi, maximum pressure sensed inline
MEAN_PRESSURE <kpsi<	float	mean pressure
STAND_DEV_PRESSURE <kpsi<	float	standard deviation for entire cable segment
MIN_KEYSTONE <degrees<	float	keystone angle in degrees, minimum sensed
MAX_KEYSTONE <degrees<	float	maximum sensed
MEAN_KEYSTONE <degrees<	float	mean value of angle for segment
STAND_DEV_KEYSTONE <degrees<	float	standard deviation of keystone values
MIN_WIDTH <inches<	float	minimum width sensed (inches)
MAX_WIDTH <inches<	float	maximum sensed
MEAN WIDTH <inches<	float	mean for segment
STAND_DEV_WIDTH <inches<	float	standard deviation of thickness values (in)
MIN_THICK <inches<	float	minimum cable thickness(inches) measured
MAX_THICK <inches<	float	
MEAN_THICK <inches<	float	
STAND_DEV_THICK <inches<	float	standard deviation of thickness values
NUMBER_MEASURED	smallint	number of measurements made in segment
CB_MECH_COMMENT	varchar(50)	some appropriate remarks

.....
.....

11.1 TABLE_COMMENTS

.....

COMPOSITE_BL_COMPONENT

There will be several rows entered for any particular billet. The number and kind will depend on the nature of the manufacturing process. The simplest would be for that of unclad superconductor rod (SC_ROD) that is stacked in, say, a hexagonal copper matrix (CU_HEX), with the front, rear, and perimeter of the stack covered (welded) with copper material (CU_NOSE, CU_TAIL, and CU_CAN, in respective order). In this case, there will be at least one row for each of the above fields, specified in, say, the above order. Each row would list, along with BUYER_ID, BL_ID_VENDOR, BL_PROD_MNEM, and the SPEC_DOC_REV_NO, the full BL_COMPONENT_NO; the latter would be different for material with different batch lots or heat numbers that contributed to the billet stack. As example, I have below a stack made from two superconductor batches, two copper lots for the matrix, a different copper lot for the periphery, another different copper lot for the nose and tail. BUYER_ID and BL_ID_VENDOR will be the same for all entries for any single billet, while BL_PROD_MNEM and SPEC_DOC_REV_NO should repeat when any component comes from more than one source. Thus we get:

BL_PROD_MNEM	BL_COMPONENT_NO
SC_ROD	IGC-9I87J-1328L
SC_ROD	IGC-7I7J-1208L
CU_HEX	WAH-9OL870D-769
CU_HEX	WAH-1OL8730D-24
CU_CAN	USD-973P-375P
CU_NOSE	JFK-1374-458F
CU_TAIL	JFK-1374-458F

(For other entries - see above.)

Note that for SC_ROD we use VENDOR_HEAT#-MACS# and for copper VENDOR_HEAT#-LOT#. We do not record what fraction of any material makes up the billet, but just identify each pertinent heat and lot number (or the conformance documents)."

.....

STRAND_RECEIVING

Only the wire that we deal with directly will appear in this table. After the initial entry here, the spools will be tracked through the SPOOL_INVENTORY_USE table.

.....

STRAND_ELEC_TEST

Tests are performed at parameter values close to those desired by the user, and extrapolated to standard (reported) settings that appear in this table. For SSC low field is 5.6 T and high is 7.0 T. Tests are specified in SSC-MAG-M-4141.

.....

STRAND_CHEM_MECH_TESTS

For the SSC, all tests are specified in appendixes of document SSC-MAG-M-4141.

.....

STRAND_INSULATION

Here we can put any information pertaining to insulation or similar treatment of special strand to be used for correction coils, etc.

.....

CABLE_PRODUCTION

This table provides information about spools of bare cable that have been manufactured and are ready for insulation. One row of entry per unique spool.

.....

CABLE_STRAND_MAPS

As cable production begins, the strand map starts with a row of entries for each spool positioned on any spool-holder of the cabling machine. Strand welds located on these spools, that join together different segments, are recorded along with appropriate original spool numbers (and segment numbers), and are entered as new rows. When cabling is completed, a final entry is made for each remaining spool of strand. (If any strand should break during cabling, cable production is halted, and a new set of spools is mounted.)

.....

CB_PROD_INVENTORY

Here we keep track of any unwrapped cable in storage and test samples sent to us from the manufacturer. We first enter the amount received or produced (as given in CABLE_PRODUCTION table). As the cable is cut and used up, we track the original spool and any progeny spools. This is done by tracking a sequential cut number (CUT_NO) and the source segment (SOURCE_NO) for that cut. First removed segment is given CB_SEG of 01. If a useful length of the original spool (00) is returned to storage, a new row is entered with this CABLE_ID and its remaining length, so that it can be recycled. If any part of the cut-off segment (01) is later returned to storage, it will also be entered as a new row with that very same CB_SEG id. If the original spool or that segment are cut again, the next progeny-segment will be labeled X2, where X is either 0 (original spool) or 1 (the first segment) etc. The cutting process can continue until a total of 35 cuts have been made (0-Z). There may be different row entries for same CABLE ID, that will differ, however, in dates of entry and feet of cable (daisy chained). The last entry with same id will give amount of cable remaining on that spool.

.....

CABLE_COATING

After a cable is manufactured, it may have to be tinned or dipped, etc., before it is wrapped. This phase can be tracked in this table.

.....

CABLE_INSULATION

After manufacture, cable is insulated. The CABLE INSULATION table has a row for each cable's insulation and wrapping step. When lumps or pinholes are sensed to be out of range, a new row is entered. If a cable must be cut during wrapping, the id of the part still on the original spool is not affected, but the cut-off piece gets a new CB_SEG (and thus new CABLE_ID as well). The id is formed as described in the CB_PROD_INVENTORY table

.....

SPOOL_INVENTORY_USE

One row for each piece of special strand or insulated cable. The entries here are initiated when some authorized individual requests special strand for coil production (or similar purpose) to be set aside, or when insulated cable is produced and is ready for storage, also when a partially used spool is returned for inventory.) We use the same tracking scheme as in CB_PROD_INVENTORY table

.....

CB_ELEC_SUMMARY

For SSC, this table contains results calculated for standard B-field values, usually 5.6 and 7.0 Tesla at 4.22 K, and the resistance-related measurements at B=0. The appropriate test specifications can be found in SSC_MAG_M_4142.

.....

CB_ELEC_MEAS

Measurements at misc. field values (usually near B=5.0, 5.6 and 6.0 Tesla), and some temperature near 4.2 K. The appropriate test specifications can be found in SSC_MAG_M_4142.

.....

CB_MAGNETIZATION

Cable magnetization measurements may be performed at different B-fields (all at 4.2 K). We allow one row per ramp-rate and field value.

.....

CB_MECHANICAL_MEAS

For SSC, this table contains results of tests stipulated in specifications in appendixes of SSC_MAG_M_4142.

.....
.....

11.2 ID_FORMATS

.....

PO_ID	varchar(24)	full identification of original purchase order, made up of the following parts, separated by hyphens:	
	BUYER_ID_MNEM	varchar(8)	buyer's mnemonic (SSC, FNAL, BNL,...)
	PO_DATE	datetime	date of issue of purchase order
	PO_NUMBER	varchar(10)	purchase order number

.....

STRAND_SPEC_ID	varchar(16)	full spec of wire type, made up of following parts, each separated by a hyphen, except for middle two; (e.g., SSC-4141-U13-C)	
	WIRE_CLASS	varchar(10)	LAB MNEM-wire classification (SSC-4141)
	ST_USE_CODE	char(1)	I, O, U use of wire
	WIRE_CU_NONCU	tinyint	10*ratio (13,15,18,...)
	WIRE_SUBTYPE	char(1)	twist/anneal properties (see the WIRE_SUBTYPES table)

.....

ST_SPOOL_ID varchar(21) this is composed from ST_PROD_ID and ST_SEG, separated by a hyphen, and fully specifies strand piece from vendor information (see STRAND_RECEIVING) (e.g., F-0688-0198-A0878-02)

ST_PROD_ID char(17) this is formed from following vendor production info: BL_ID_VENDOR and SPOOL_NUMBER, NOT separated by hyphen

BL_ID_VENDOR char(13) billet code number assigned by vendor (from COMPOSITE_BILLET table): VENDOR_ID_CHAR, VENDOR_JOB_NO, MFG_BL_NO, BILLET_PART, each separated by a hyphen;specifies up to spool

VENDOR_ID_CHAR char(1) in ORGANIZATION_MNEMS table

VENDOR_JOB_NO smallint in COMPOSITE_BILLET table

MFG_BL_NO smallint in COMPOSITE_BILLET table

BILLET_PART char(1) in COMPOSITE_BILLET table

SPOOL_NUMBER smallint spool number (see STRAND_PRODUCTION table)

ST_SEG varchar(3) strand segment (see STRAND_RECEIVING table)

.....

BUYER_SPOOL_ID varchar(17) equivalent to our CABLE_ID, formed from BUYER-ID-MNEM, ST_USE_CODE-VENDOR_ID_CODE-ST_SERIAL_NO-ST_SEG, all parts separated by hyphen (e.g., SSC-O-F-03889-02)

.....

CABLE_SPEC_ID varchar(16) formed from CABLE_CLASS, CB_USE_CODE, and NO_STRANDS (e.g., SSC-4142-I23); the Cu/NonCu can be traced (see CABLE_TYPE_SPEC table)

.....

CABLE_ID varchar(17) formed from CABLE_PROD_ID, and CB_SEG, separated by a hyphen: SSC-I-F-00123-I8

CABLE_PROD_ID char(13) formed from CB_BUYER_MNEM,CB_USE_CODE, CB_VENDOR_ID_CHAR, and CB_SERIAL_NUMBER, all separated by a hyphen: SSC-I-F-00123

CB_BUYER_MNEM varchar(8) mnemonic for buyer (SSC,BNL,FNL,DSY...)

CB_USE_CODE char(1) one character stipulating use of cable

CB_VENDOR_ID_CHAR char(1) one character for id of manufacturer

CB_SERIAL_NUMBER int serial number assigned to any cable/strand

CB_SEG varchar(3) segment of cable: END or BEG for test pieces; or a two character object that is specified by the sequential cut of any cable and the source segment (0-Z) (see explanation in CB_PROD_INVENTORY

table for more)

CB_USE_CODE char(1) — I=inner dipole, O=outer dipole, Q=quadrupole, with 0-9, and rest of letters available for future use; the ST_CB_USE_MNEMS table contains all the current id's corresponding to cable and strand use codes and mnemonics

---

CB_VENDOR char(1) — L=LBL, P=Supercon, I=IGC, X=OST, F=Furukawa, with 0-9, and rest of letters available for future use; the ORGANIZATION_MNEMS table contains all current id's corresponding to lab and vendor id's and mnemonics

---

COIL_ID char(10) — formed from CL_PREFIX and CL_SER_NO; when cancatenated for use by SSC, the parts will be separated by a hyphen; when there are less than five characters in the serial number (e.g., for BNL), enough leading zeros will be added to comply with the overall SSC convention

- CL_PREFIX char(4) — composed from CL_TYPE,CL_SYSTEM,CL_SERIES, and CL_PLACE, all strung together without spaces
  - CL_TYPE char(1) — coil (magnet) type; D=dipole,Q=quad, etc
  - CL_SYSTEM char(1) — particular accelerator or sub-system: R=RHIC, S=SSC, T=Tevatron, etc
  - CL_SERIES char(1) — design series for that coil/magnet type
  - CL_PLACE char(1) — location of coils on magnet; e.g., I for inner, O for outer, C for no distinction
- CL_SER_NO varchar(5) — serial number for coil; leading zeroes may be added at time of cancatenation (for SSC)

Current examples:

SSC dipole inner	DDI-0033	would become DSDI-00033
SSC short dipole	DSSI-0013	becomes DSSI-00013
RHIC dipole	DRAC-0011	becomes DRAC-00011
RHIC quad	QRAC-0001	becomes QRAC-00001
SSC	LLNI-12	would be DSDI-00012
SSC	DD11-I-4	would need a new serial number.
LBL dipole	D14-A3-L1	needs new series and serial number!

(for LBL: D=dipole, series=14, serial no=A3, layer-1=L1)

MAGNET_ID char(9) — formed from MG_PREFIX and MG_SER_NO; when cancatenated for use by SSC, the parts will be separated by a hyphen; when there are less than five characters in the serial number (e.g., for BNL), enough leading zeros will be added to comply with the overall SSC convention

- MG_PREFIX char(3) — composed from MG_TYPE,MG_SYSTEM,MG_SERIES, all strung together without space or hyphen
  - MG_TYPE char(1) — magnet type; e.g., D=dipole, Q=quad, etc

```
                MG_SYSTEM    char(1)     particular accelerator or sub-system:
                                         R=RHIC, S=SSC, T=Tevatron, etc

                MG_SERIES    char(1)     design series for that magnet type

          MG_SER_NO     varchar(5)    serial number for magnet; leading zeroes may
                                      be added at time of cancatenation (for SSC)

Current examples:
  SSC dipole magnet  DD-0012   would be DSD-00012
  SSC short dipole   DSS-0008  would be DSS-00008
  RHIC quad          QRA-0001  would be QRA-00001
  LBL dipole           ??                DSL-00001  ?
  LBL dipole           D14-A3  needs new serial number and series code

.....................................................................................
#####################################################################################
                                      END!
```