ALMA Band 6 Cartridge

Database Design

2004-12-14

Version 1.3

Revisions

Table 1: Document Revisions

Revision Number	Date	Who	Details
1.0	2002-10-17	jee	Initial
1.1	2002-10-18	jee	Modified schema diagram after creating tables in SQL Server
1.2	2003-05-08	jee	Modified schema diagram
1.3	2004-12-14	jee	Updated Schema

Contents

List of Tables

1. Introduction

This memo provides details of the database structure to hold test data for both ALMA cartridges and SIS mixers. The general database structure, future work, and details of the design schema are presented here.

2. General Database Structure

The specific data to be stored in the database is apparent from the field names in the schema diagram (Figure 5) and is not repeated here. Relationships between the tables required for each type of test are not as obvious, so that information is given in this section. Details of each table are provided following the section entitled "Structure of Tables"

Each cartridge, identified by a cartridge ID stored in table `Carts`, can have multiple, time tagged notes that are stored in table `NotesCart`. Table `CartComps` provides a general way to store information about the components used in the cartridge, and each component entry can have multiple notes stored in table `NotesCartComps`. This general design paradigm is repeated for mixers and preamplifiers. Mapping tables `MxrCartMaps` and `MxrAmpMaps` provide the means to document each time mixer-preamps are inserted and removed from The `DateRemoved` field in table `CartComps` identifies those cartridge components that have been replaced by other components. The `DateRemoved` field is blank unless the corresponding component has been removed from the cartridge.

When a cartridge fails a particular test, it is repaired and then retested. Information related to retesting is stored in table `CartTest` and its associated notes table `NotesCartTest`. Table `CartTest` provides a way of uniquely identifying each test run on the cartridge.

Mixers installed in each cartridge are documented in table `MxrID` along with notes about the mixer, which are documented in table `NotesMxrID`. Table `MxrID` provides a way of identifying when mixers are removed and replaced in the cartridge using the fields `DateInstalled` and `DateRemoved`.

The same database is used to store both stand-alone mixer and cartridge information. Cartridge information is always referenced to the highest level table, `Carts`. Mixer performance, measured prior to installation in the cartridge using the mixer test set, is not initially related to any particular cartridge, so mixer data does not require any of the tables specifically related to cartridges. The database design permits mixer data to be entered without referring to a cartridge by allowing entries into the database starting with table `MxrID`.

Mixers, like cartridges, may need rework after failing either a single test or a series of test, and measurements will need to be repeated after any rework. Documenting the rework in table `MxrTest` and its associated notes table `NotesMxrTest` is useful for identifying multiple entries of the same type of test.

The type of measurement is stored in table `MeasType` and includes measurements such as:

1) Pumped and unpumped I-V curves,
2) unpumped I-V curves *vs*. magnet current,
3) total power as a function of bias voltage,
4) receiver noise temperature *vs*.
   i) IF,
   ii) LO frequency,
   iii) LO power, and
5) sideband ratio *vs*. IF and LO frequency.

LO frequencies are stored on table `LOFreq`, and LO power and magnetic field current are stored in child tables to `LOFreq`.

3. Structure of Tables

Figure 4 shows the database schema that holds information about the configuration of the cartridges, mixers, and preamps. This includes information such as what mixers are installed in each cartridge, what preamps are installed on mixers, and what components are installed in each mixer, preamp, and cartridge.

Figure 5 shows the database schema that contains test data for the mixers, cartridges, and preamps.

Each table is described in the following sections.

3.1 Cartridges

3.1.1 Tables Carts and NotesCart

Shown in Figure 4, table `Carts` is the highest-level table and contains the ID for the cartridge, which can be as simple as a string of characters, or may include intelligence[1], such as the cartridge band and date of manufacture. Table `Carts` is structured so that there are no uniqueness requirements for the cartridge ID, because the database and child tables use the key field `keyCarts`, which is automatically generated, to maintain proper relationships. There is one record in `Carts` for each cartridge.

Notes about each cartridge are held in the table `NotesCart`. The one-to-many relationship of `NotesCart` with respect to `Carts` means that multiple notes can exist for each cartridge. The field `PhotoURL` in table `NotesCart` holds the URL to files containing photos of the cartridge. Although it is possible to store the photo in the database using a binary field format, Microsoft recommends against this to maintain optimum database responsiveness.

3.1.2 Tables CartComps and NotesCartComps

Table `CartComps`, shown in Figure 4, contains information about the components installed in the cartridge. This table is general and can include information about OMT's, warm IF amplifiers, LO chains, bias cards, M&C cards, *etc*. Information about which mixers are installed in the cartridge is stored in the `Mxrs` table discussed below.

Multiple, time-stamped notes associated with each component mounted in the cartridge are stored in the table `NotesCartComps`. The one-to-many relationship of `NotesCartComps` with respect to `CartComps` means that multiple notes can exist for each cartridge component. The field `PhotoURL` in table `NotesCartComps` holds the URL to files containing photos of the cartridge components.

3.1.3 Table MxrCartMaps

Table `MxrCartMaps` provides the mapping between cartridges and the mixers installed in each cartridge. The separate table structure provides a way for the same mixer to be installed and removed for a cartridge an indefinite number of times. Each time a mixer is installed in a cartridge, a new record is created in `MxrCartMaps`. When that

[1] Adding intelligence to ID numbers is to be strongly discouraged, because later changes to the ID number format can result in extensive and time consuming updates to historical data stored in paper format. It is much better to simply use the database key value as the ID number and use database queries to obtain all information about the cartridge.

particular mixer is removed, the DateRemoved field changes from null to the date that the device was removed. This provides a means to track when mixers are added and removed from cartridges.

3.2 Mixers

3.2.1 Table Mxrs and NotesMxrs

Table Mxrs contains the ID string[2] about a particular mixer-preamp pair as assigned in the mixer assembly lab.

Notes about each mixer are held in the table NotesMxrID. The one-to-many relationship of NotesMxrID with respect to table MxrID means that multiple notes records can exist for each cartridge test.

3.2.2 Table MxrAmpMaps

Table MxrAmpMaps provides the mapping between mixers and the two preamplifiers installed on each mixer. The separate table structure provides a way for the same preamplifiers to be installed and removed for a mixer an indefinite number of times. Each time a preamplifier is installed in a mixer, a new record is created in MxrAmpMaps. When that particular preamplifier is removed, the DateRemoved field changes from null to the date that the device was removed. This provides a means to track when preamplifiers are added and removed from mixers.

3.3 Preamplifiers

3.3.1 Table Preamps and NotesPreamps

Table Preamps contains the ID string about a particular preamplifier as assigned during preamp assembly.

Notes about each preamp are held in the table NotesPreamps. The one-to-many relationship of NotesPreamps with respect to table Preamps means that multiple notes records can exist for each preamp.

3.4 Measurement Data

3.4.1 Table CartTest and NotesCartTest

Table CartTest contains descriptions of the tests performed on the cartridge. The purpose of this table is to document when multiple tests, perhaps of the same type, are run on a particular cartridge. For example, if the cartridge fails its DC continuity test, then something will require repair and the test will be rerun. In that case, two records are required in CartTest for both the initial test and for the retest. To provide implementation flexibility, CartTest contains some redundancy between the fields TestType, which is numeric, and Description, which is a text-based field.

Notes about each cartridge test are held in the table NotesCartTest. The one-to-many relationship of NotesCartTest with respect to table CartTest means that multiple notes records can exist for each cartridge test.

[2] Mixer ID strings are defined in http://www.cv.nrao.edu/~jeffland/MixerNumbering.pdf

3.4.2 Table MxrTest and NotesMxrTest

Information about mixers that require rework is stored in the table `MxrTest`. This table provides a way to distinguish multiple sets of the same types of mixer measurements for the same mixer. Since mixers that have been reworked will retain the same ID, table `MxrTest` provides information to uniquely identify the purpose of subsequent sets of mixer measurements.

For example, mixer/preamp UVA-1 may have two records in table `MxrTest`: the original test and a retest after replacing a cracked mixer chip. Each test could have the same type of IV and noise temperature records, as stored in table `MeasType` and described below.

Notes about each mixer test are held in the table `NotesMxrTest`. The one-to-many relationship of `NotesMxrTest` with respect to table `MxrTest` means that multiple notes records can exist for each mixer test.

3.4.3 Table MeasType

Table `MeasType` stores the type of measurement, such as I-V curves, noise temperatures, and sideband ratios. Mixers may have multiple records for the same type of measurement, and table `MxrTst`, described previously, provides a way to uniquely identify each measurement type.

Other types of measurements include:

1. Total receiver power *vs*. LO power
2. Total receiver power *vs*. magnet current
3. Total receiver power *vs*. IF
4. Total receiver power *vs*. mixer bias

3.4.4 Table TempPress

Dewar physical temperatures and pressures as well as RF hot and cold load physical temperatures are stored in table `TempPress`. This table is linked by the foreign key field `fkMxrTest` to mixer test table `MxrTst` to provide a way to relate multiple Dewar cool-down temperature/pressure records to each new mixer test.

3.4.5 Table IFSys

The IF system parameters of interest include IF hot and cold load temperatures, total noise power when the IF system input is connected to the IF hot and cold loads, and a mapping of the desired IF attenuator value to the frequency of the IF system. IF noise temperatures, which is determined from IF total power measurements stored in this table, usually change slowly with time. This means that the noise performance of the IF system can be measured just a few times during the measurement of receiver noise temperature. Consequently, the IF parameters are linked to the measurement type table `MeasType` through the foreign key field `fkMeasType`. This allows IF parameter data to be updated at a rate that's independent from the measurement of the other parameters.

3.4.6 Table LOFreq

The local oscillator frequency, in GHz is stored in this table. The optimum LO power for a particular LO frequency is stored in field `LOPwrOpt` in this table.

3.4.7 Table LOPwr

Table `LOPwr` holds local oscillator powers for each particular LO frequency. The units for LO power have been arbitrarily set to dBm. This table contains fields to store both LO power level and the value of the voltage that controls the LO power. Total noise power is stored in the child table `TPwr` and linked to table `LOPwr` *via* the foreign key field `fkTPwr`. LO frequency is stored in the parent table, so that multiple LO power levels (and the total power measurements with the link to `TPwr`) can be stored for each particular LO frequency.

3.4.8 Table IMag

The mixer's magnetic field current is stored in table `IMag`. Magnetic field current is usually optimized by minimizing the Josephson spikes in the un-pumped I-V curves, and hence seems to be independent, or at least weakly dependent, on LO power or frequency. Nevertheless, the table `LOFreq` is assigned as its parent in case it is found that the optimum magnetic field varies with LO frequency. No child tables are assigned to table `IMag` because, based on empirical evidence from numerous mixer measurements, the optimum magnetic field current appears to be independent of LO power, IF frequency, and mixer bias.

3.4.9 Table FreqIF

This table simply stores the intermediate frequency of the measurement. Total power data, which is frequently recorded as a function of IF, is stored in the child table `TPwr`.

3.4.10 Table TPwr

This table maps a record from the relevant parent table to a single total receiver noise power record. Because the structure of table `TPwr` is common for a number of parent tables, moving the common total power fields to a separate table reduces redundancy.

The following are examples of using `TPwr`. Assume we want to measure and record receiver noise temperature as a function of LO power at a fixed bias voltage. As shown in Figure 1, LO power is stored in table `LOPwr` while total receiver noise power corresponding to each LO power level is stored in table `TPwr` through the linkage provided by the foreign key field `fkTPwr`.

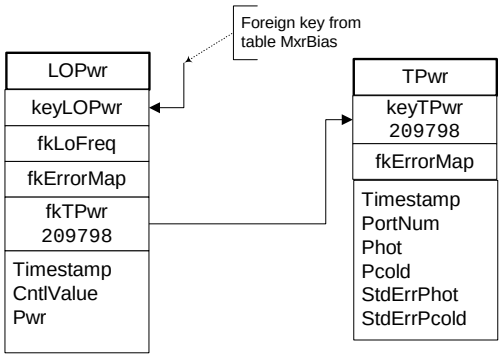

Figure 1: Use of table `TPwr` with table `LOPwr`

For another example, assume we want to store noise powers as a function of IF, with LO frequency, LO power, bias voltage, bias current, and magnet current fixed. As drawn in Figure 2, foreign key field `fkTPwr` in table `FreqIF` holds the key value from table `TPwr` so the total power record is linked uniquely to the IF table record.

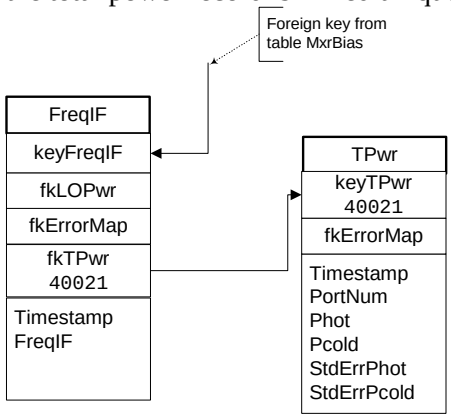

Figure 2: Use of table `TPwr` with table `FreqIF`

3.4.11 Table MxrBias

Table `MxrBias` stores junction voltage (using field `Vj`) and junction current (field `Ij`) for a particular component mixer (stored in the field `CompMxrNum`).

Pumped and unpumped IV curves are a function of LO frequency and magnet current, so table `MxrBias` contains a foreign key linking multiple `MxrBias` records to a single LO power record stored in `LOPwr`. The magnetic field current for a particular mixer bias is stored in table `IMag`, which is linked to table `LOFreq`.

3.4.12 Table Errors and ErrorMap

Table `Errors` and `ErrorMap` provide a mechanism to store multiple error messages that refer to the measurement of a particular parameter, where the parameter may be stored in a number of different tables. Each parent data table holding parameter values includes a foreign key field `fkErrorMap` that refers to table `ErrorMap`, where `ErrorMap` simply contains a unique key. Multiple error messages related to a single measurement record are easily accommodated by the database because, as shown in Figure 3, more than one record in table `Errors` can hold the foreign key value that refers to table `fkErrorMap`.

Table `ErrorMap` is essential, because it insures unique key values. If the foreign keys in `Errors` held key values of the parent data tables, there might be multiple values of the same foreign keys in `ErrorMap`, because key values in different parent tables are not necessarily unique.

Generic number fields `Number1` and `Number2` provide a means to store unspecified error conditions.

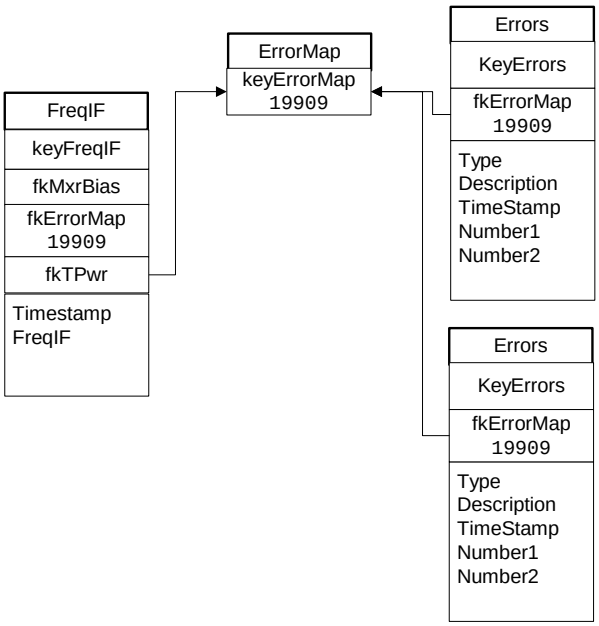

Figure 3: Use of Tables `Errors` and `ErrorMap`

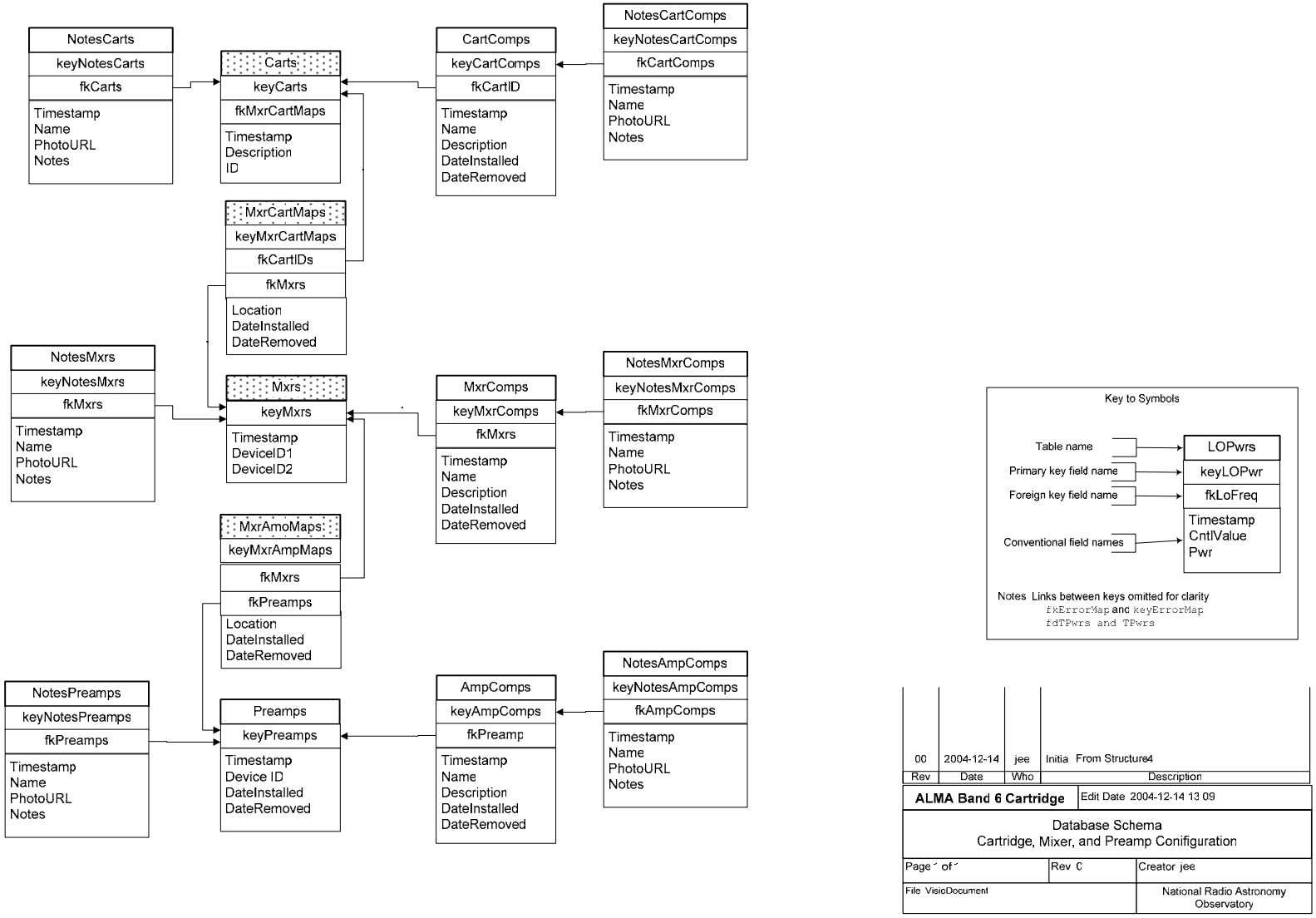

Figure 4: Database Schema, Cartridge, Mixer, and Preamp Configuration (Figure embedded in document)

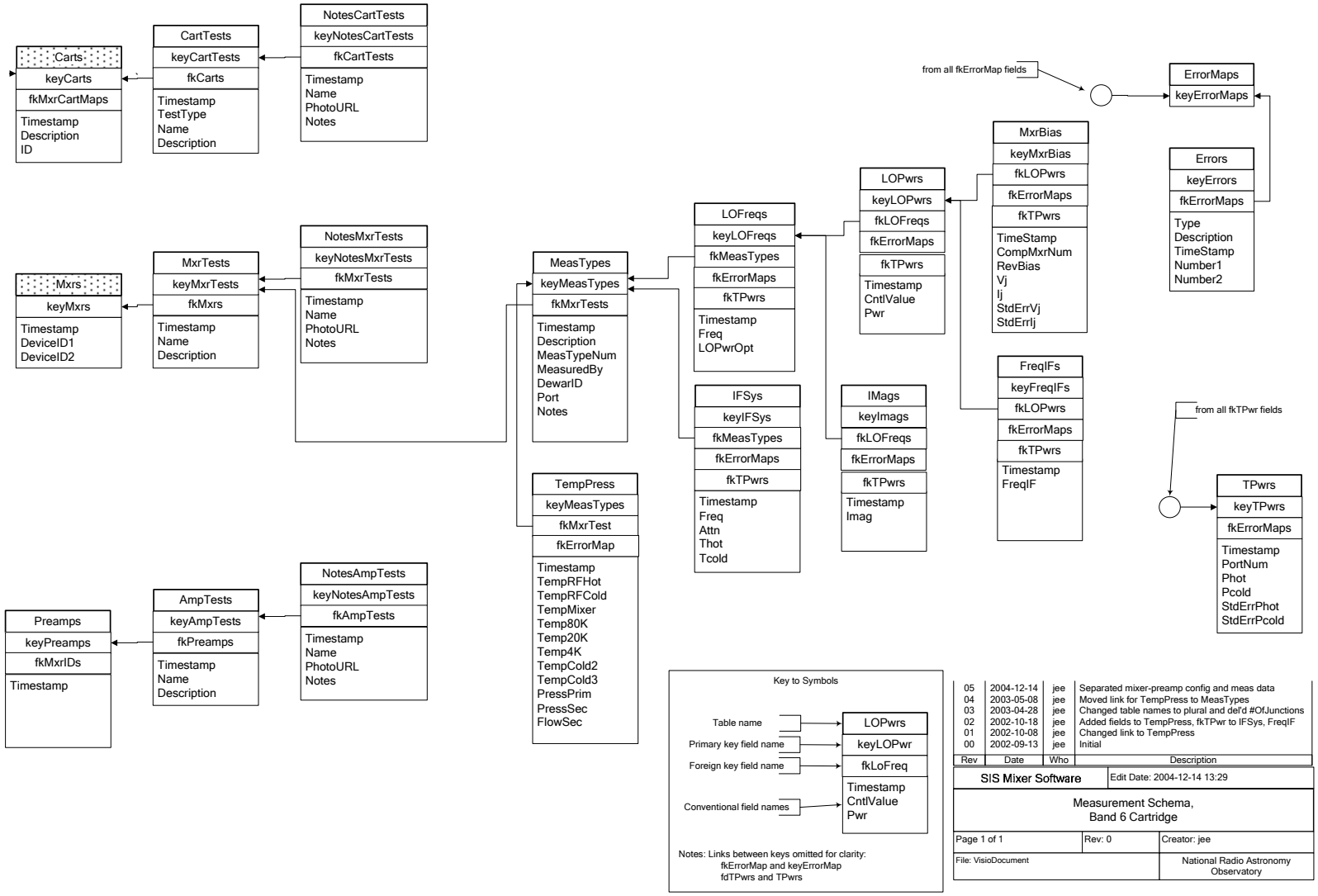

Figure 5: Database Schema, Measurements
(Figure embedded in document)