



LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)
Data Management

LSST Database Baseline Schema

Colin Slater, Fritz Mueller, Brian Van Klaveren, Jacek Becla

LDM-153

Latest Revision: 2020-07-29

Draft Revision NOT YET Approved - This LSST document has been approved as a Content-Controlled Document by the LSST DM Change Control Board. If this document is changed or superseded, the new document will retain the Handle designation shown above. The control is on the most recent digital document with this Handle in the LSST digital archive and not printed versions. Additional information may be found in the corresponding DM RFC. - **Draft Revision NOT YET Approved**

Change Record

Version	Date	Description	Owner name
1	2007-01-12	Initial Version.	Jacek Becla
2	2007-02-14	General edits.	Jacek Becla
3	2007-02-21	General edits.	Jacek Becla
4	2007-02-27	General edits.	Jacek Becla
5	2007-03-05	General edits.	Jacek Becla
6	2007-03-07	General edits.	Jacek Becla
7	2007-04-09	General edits.	Jacek Becla
8	2007-09-05	General edits.	Jacek Becla
9	2011-07-04	General edits.	Jacek Becla
10	2011-07-14	General edits.	Jacek Becla
11	2013-08-02	Major refresh of the schema to reflect science requirements described in Data Products doc. Added sections about schema evolution, provenance	Jacek Becla, Kian-Tat Lim
12	2013-09-30	Resynchronized with the latest Data Products Definition Document (v Sep 23 2013) and corresponding schema.	Jacek Becla
13	2013-10-10	TCT approved	R. Allsman
14	2013-10-18	Fixed Table of Contents	Jacek Becla
15	2019-06-27	Switch to yaml-defined schema	Colin Slater



Contents

1 Schema Format and Documentation	1
2 Stored Procedures and Functions	1
3 Selected Catalogs	2
3.1 Core Catalogs	2
3.1.1 Sizes	3
3.1.2 Partitioning	4
3.1.3 Diagram	5
3.2 Metadata Catalogs	6
3.2.1 Sizes	6
3.2.2 Diagram	6
3.3 Provenance	8
3.3.1 Size	8
3.3.2 Diagrams	8
3.4 Other Tables	9
4 Selected Design Aspects	9
4.1 Provenance	12
4.2 Schema Evolution	14
4.2.1 Released Data Sets	14
4.2.2 Level 1 (Up-to-date Catalog)	15
5 Naming Conventions	15
6 Conventions for NULL and missing data	16
7 Baseline Schema	17
7.1 ZZZ_Schema_Description	18
7.2 DiaObject	18
7.3 SSOject	23



7.4	DiaSource	26
7.5	DiaForcedSource	34
7.6	DiaObject_To_Object_Match	35
7.7	Object	35
7.8	Object_Extra	52
7.9	Object_APMean	53
7.10	Object_Periodic	53
7.11	Object_NonPeriodic	54
7.12	Source	54
7.13	Source_APMean	57
7.14	ForcedSource	57
7.15	RawAmpExposure	58
7.16	RawAmpExposureMetadata	59
7.17	RawCcdExposure	59
7.18	RawCcdExposureMetadata	61
7.19	RawExposure	61
7.20	CcdVisit	62
7.21	CcdVisitMetadata	64
7.22	Visit	64
7.23	VisitMetadata	66
7.24	Visit_To_RawExposure	66
7.25	prv_ProcHistory	66
7.26	prv_Pipeline	67
7.27	prv_cnf_Pipeline	67
7.28	prv_Task	68
7.29	prv_cnf_Pipeline_Tasks	68
7.30	prv_cnf_Task	68
7.31	prv_cnf_Task_Columns	69
7.32	prv_cnf_Task_Files	69
7.33	prv_cnf_Task_KVParams	70



7.34 prv_Node 70

7.35 prv_cnf_Node 70

7.36 prv_DataBlock 71

7.37 prv_RowIdToDataBlock 71

7.38 prv_TaskExecution 72

7.39 prv_TaskExecutionToInputDataBlock 72

7.40 prv_TaskExecutionToOutputDataBlock 73

7.41 sdqa_ImageStatus 73

7.42 sdqa_Metric 74

7.43 sdqa_Rating_ForAmpVisit 74

7.44 sdqa_Rating_CcdVisit 75

7.45 sdqa_Threshold 75

7.46 LeapSeconds 76

7.47 ApertureBins 76

Draft

LSST Database Baseline Schema

1 Schema Format and Documentation

The LSST Database schema master files are stored in yaml files which are designed to be read by the “Felis”¹ schema manipulation tool. The files live in the `sdm_schemas`² package and are version-controlled via git together with the rest of the LSST software.

The yaml files are designed to contain both the required attributes necessary for instantiating a database table and extra per-column metadata for enabling a richer description of the data, such as additional descriptive text, physical units, or IVOA UCDs. The choice of a yaml format also enables additional metadata attributes to be added as the need arises during development.

The schema tables presented in this document are automatically generated from the yaml files in the `sdm_schemas` package.

2 Stored Procedures and Functions

We maintain a small set of stored procedures and functions, as well as a set of user-defined functions (UDFs). Since stored procedures and functions typically are vendor-specific, we have tried to limit them to the bare minimum; nevertheless, we now have ~ 10 such procedures, most related to converting different time formats. The most frequently used functions are written as native user-defined functions for performance and portability reasons. They include support for spherical geometry, photometry conversions, and statistical functions (e.g., median). These spatial functions rely on custom HTM indices to allow fast spatial selections, such as point in circle, polygon or ellipse searches. We are collaborating with the team from Johns Hopkins University that implemented HTM indexing and UDFs for SDSS, and we already incorporated many of their lessons learned into our UDFs (their UDFs were written in C# for SQL Server, since then the SDSS team converted them to C++ using automated conversion tools.)

¹<https://github.com/lst-dm/felis/>

²https://github.com/lst/sdm_schemas/

Since these UDFs are not LSST-specific and may be of broader interest (to MySQL community and beyond), we maintain them as an open-source project independent from LSST. This project, SciSQL, is hosted at <https://github.com/smonkewitz/scisql>.

3 Selected Catalogs

The baseline schema is driven primarily by the Data Products Definitions Document (DPDD). The schema can be divided into several logical groups:

- Core Catalogs
- Metadata Catalogs
- Provenance Catalogs
- SDQA

Here, a catalog refers to a group of closely-related tables that act as a unified dataset from an end user's point of view.

3.1 Core Catalogs

The core catalogs can be divided into:

- Level 1 Catalogs: DiaObject, SSOBJECT, DiaSource and DiaForcedSource.
- Level 2 Catalogs: Object, Source, ForcedSource.

DiaObject table contains information about astronomical objects measured on difference images. For reproducibility reasons (described in LDM-135, chapter 3.1), we expect to create a new row each time there is an update to a diaObject, thus the total number of rows in that table is higher than the unique number of diaObjects.

SSObject table contains information about Solar System (also called moving) objects.

DiaSource table contains information about individual measurements of astronomical objects on different images.

DiaForcedSource contains photometry measurements about low signal-to-noise detections done on individual difference image exposures in each place where an object was detected on a previous difference image within one month.

Object Catalog contains information about static astronomical objects measured on a stacked image. It is the most commonly used catalog.

Source Catalog contains information about high signal-to-noise (>5 sigma) detections on single frame images.

ForcedSource table contains photometry measurements about low signal-to-noise detections done on individual exposures in each place where an object was detected on a stacked image. Due to the nature of these detections only a few parameters can be measured, thus the table is very narrow, however it is by far the largest in terms of row-count.

DiaObject_To_Object_Match table keeps a mapping of diaObjects to nearby objects, used primarily for user data analysis. For rapid real-time queries we maintain three nearbyObj columns in DiaObject.

3.1.1 Sizes

The table below provides a rough count (size, rows, columns) for core tables (in the last Data Release, counting data only, e.g., size of persistent overheads such as indices is not reflected here). The exact numbers for each DR can be found in LDM-141.

Table	Size [TB]	Rows [billions]	Columns
Object (narrow)	~ 107	~ 47	~ 330
Object (all extras)	~ 1,200	Largest ~ 1,500	~ 7,650
Source	~ 5,000	~ 9,000	~ 50
ForcedSource	~ 1,900	~ 50,000	6
SSObject	~ 0.003	0.006	~ 80
DiaObject	~ 27 (76 max)	~ 15 unique (~ 44 max)	~ 260
DiaSource	~ 23	~ 45	~ 70
DiaForcedSource	~ 13	~ 300	8

We expect to keep refining the baseline schema extensively through LSST's construction, and will likely end up adding a small number of additional columns we haven't thought about so far. Since the schema is used for cost estimation, such changes would increase overall size of the database (and consequently, the cost), so we have reserved in the costing model (LDM-141) 25% of each table size³ for yet-unknown columns. Further, the costing model assumes a 3% increase in row size for each Data Release.

3.1.2 Partitioning

To optimize disk I/O and ensure frequently accessed columns are colocated, and to avoid unnecessary I/O, selected Catalogs are *vertically partitioned*.

The Object Catalog is partitioned vertically into the following tables:

- Object — a table with ~330 most frequently used columns. This table is expected to contain ~47 billion rows by the end of the survey, containing ~48% in the first data release.
- Object_APMean — a very narrow table containing aperture photometry mean values, on average ~8 per object.
- Object_Periodic — a very narrow table containing definitions of periodic features, on average ~32 per object.
- Object_NonPeriodic — a very narrow table containing definitions of non-periodic features, on average ~20 per object.

³With the exception of ForcedSource and DiaForcedSource

- Object_Extra — a very wide table with remaining, less frequently used columns, such as covariances for 2 models (Point Source and Bulge+Disk), and Bulge+Disk samples. These are packed into 3 blobs, 66, 168, and 7,200 elements in each blob respectively. Packing is driven by expected access pattern: entire set of elements in a blob or nothing will be accessed. There will be one such row for each row in the Object table.

The Source Catalog consists of two tables: Source and Source_APMean, the latter is similar to Object_APMean.

Since managing trillions of rows in a single table is close to impossible, in addition to vertical partitioning, all core catalogs will be horizontally partitioned, The Level 1 catalogs will be partitioned by time into a small set of tables, and all large Level 2 catalogs will be partitioned spatially into ~20,000 partitions for each table. Further details related to horizontal partitioning of these tables are discussed in LDM-135 (*LSST Database Design*).

3.1.3 Diagram

The diagram below depicts all core tables, and their relations.

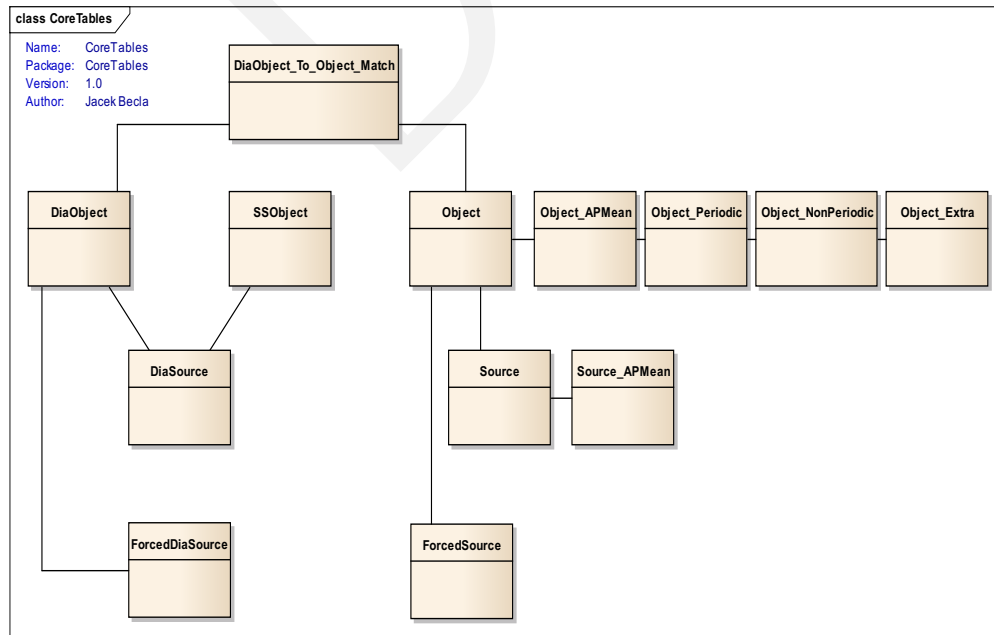


FIGURE 1: Core table relationships

3.2 Metadata Catalogs

Metadata catalogs contains information about LSST “raw exposures” and “science calibrated visits”. Information is tracked at different levels:

- focal plane (entire exposure/visit),
- raft (raft = 9 ccds)
- ccd (there are 189 ccds per focal plane)
- amp (there are 16 amplifiers per ccd).

When appropriate, we denormalize information into “lower-level” table (eg, information from ccd-level is repeated for each amp) to avoid extraneous joins for common accesses.

In addition to fully structured tables, for selected tables we expect to maintain an additional table with key-value pairs. This will allow us to introduce additional “columns” without altering corresponding table’s schema (at the cost of degraded performance). This might be particularly handy when trying to determine what information would be particularly useful to derive from the existing columns. Periodically, most commonly used key-value pairs will be converted into regular columns.

In addition, we maintain a table that maps raw exposures to visits. Typically it will contain two rows for each visit.

3.2.1 Sizes

Size of the metadata tables is not posing any major challenge. LSST is expected to produce ~one thousand visits per night, which leads to ~3 million visits during 10 year lifetime of the survey.

3.2.2 Diagram

The diagram below depicts the metadata tables, and their relations.

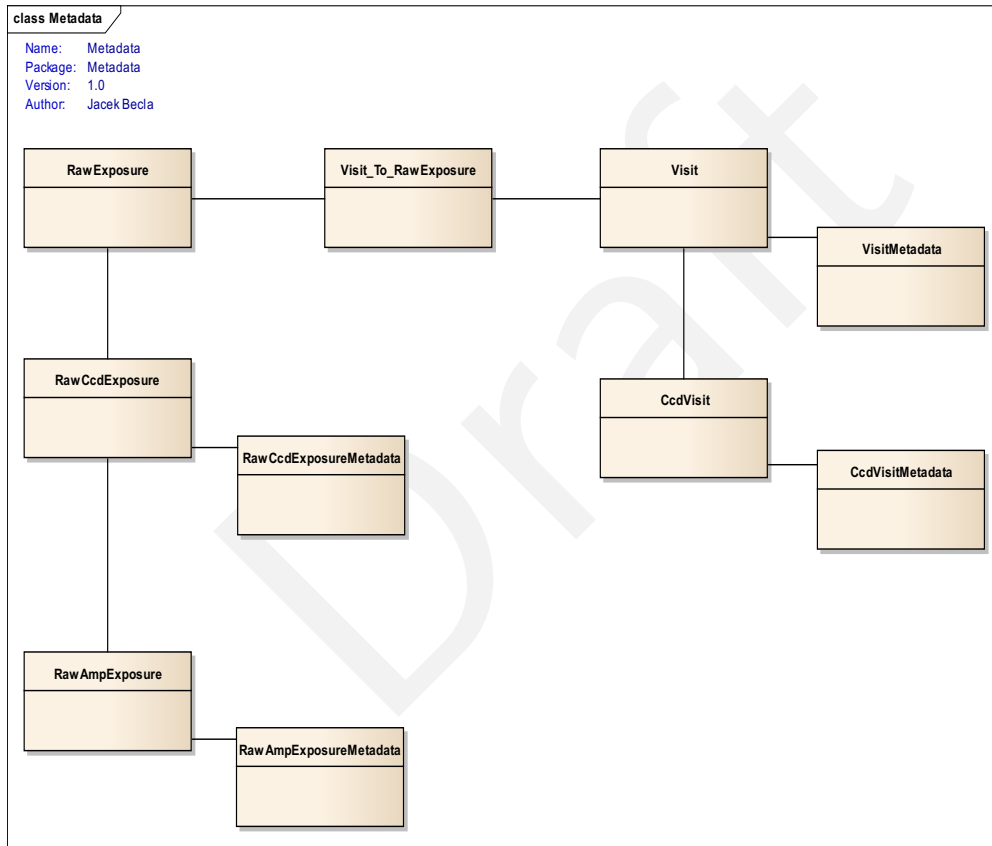


FIGURE 2: Metadata table relationships

3.3 Provenance

In LSST, we define Provenance as execution trace, or in other words information “how we came up with the data”. Provenance will be used primarily:

- to help with data QA, for example to detect which parts of data were affected by a faulty algorithm or a bad node,
- as a recipe how to regenerate intermediate results.

Specifically, Provenance is responsible for capturing configuration of:

- software (which pipelines were run on given data, what processing stages were part of each pipeline, what versions of algorithms were used, how was processing parallelized and on which computing node each piece was executed etc) and
- hardware (what was the configuration of focal plane, each raft, ccd, amplifier, filter etc).

3.3.1 Size

In terms of size, we expect provenance to be a very small fraction of the entire data set. The largest contribution is an 8-byte column we are adding for every core tables (with the exception of DiaForcedSource table, as it will have the same provenance as its corresponding DiaSource table), and selected metadata tables. Note that in some cases, in particular for the very narrow ForcedSource table with trillions of rows, that contribution is non-negligible.

3.3.2 Diagrams

The diagram below depicts the metadata tables, and their relations. Note that there is a separate section discussing provenance design in details, see below *Selected Design Aspects* below.

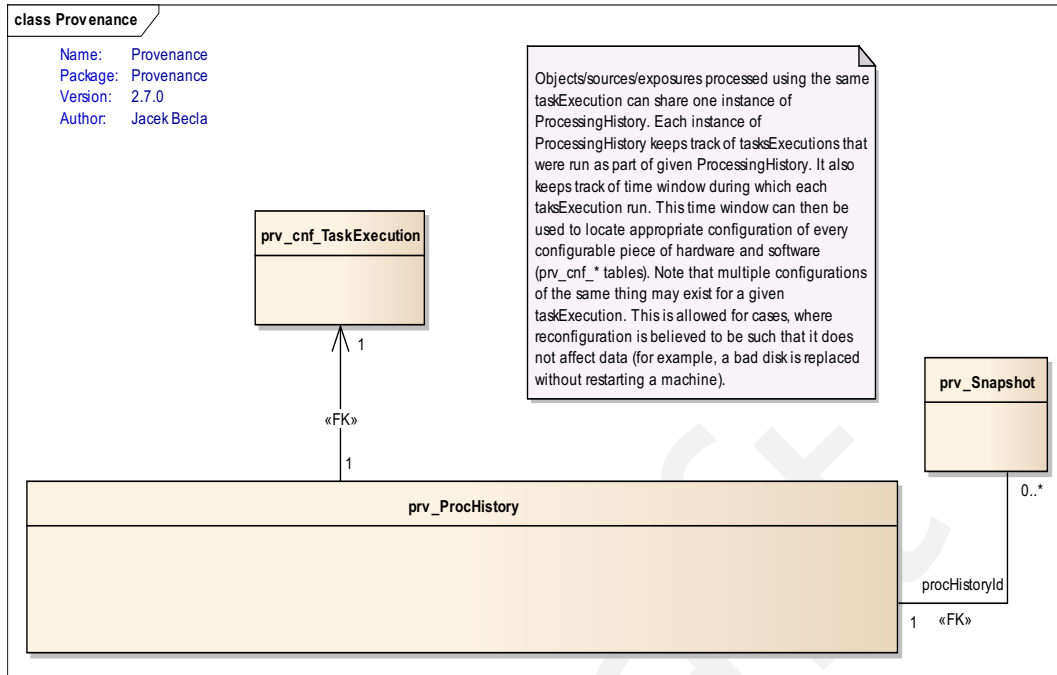


FIGURE 3: Provenance table relationships

3.4 Other Tables

In addition to the catalogs/tables described above, there are several groups of other tables in the baseline schema. In terms of size and complexity they pose very little challenge comparing to the catalogs described above, thus this document does not focus on discussing these tables (and we have left designing details of these tables for closer-to-commissioning). They include Software Data quality Assurance tables (prefixed with sdqa_), used for tracking various quality aspects of the LSST catalogs.

4 Selected Design Aspects

Below we covered in details selected, most interesting aspects of the schema design.

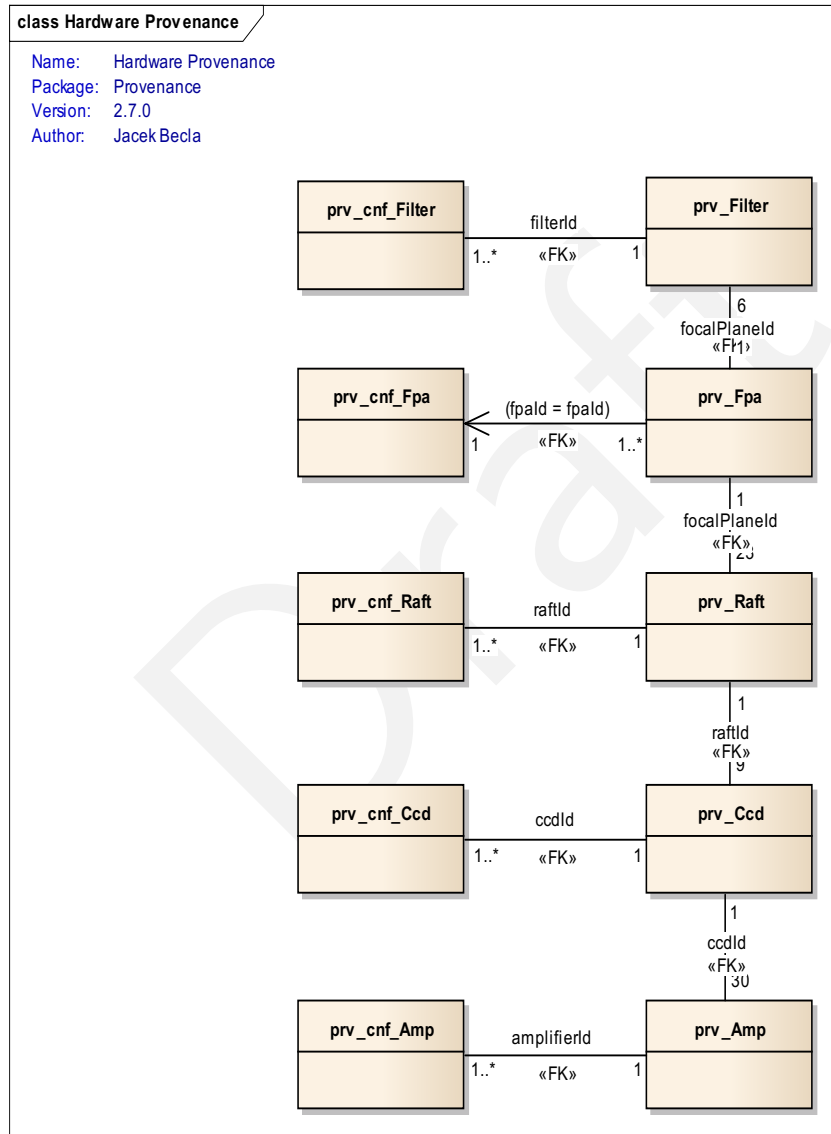


FIGURE 4: Hardware provenance table relationships

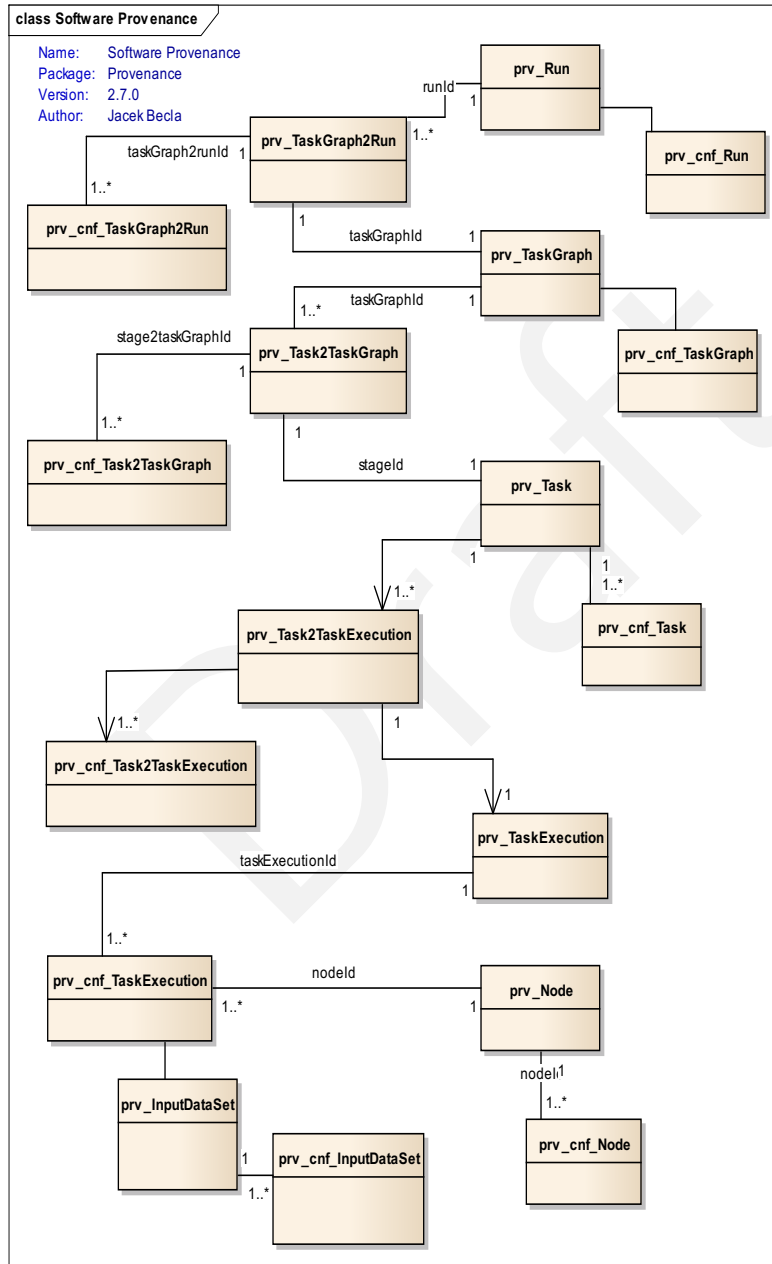


FIGURE 5: Software provenance table relationships

4.1 Provenance

Provenance is fundamentally concerned with two things: recording the state of relevant portions of the system and associating state with the generation of an output from inputs. Given the size of data, provenance information cannot be captured per data element. Instead data elements with the same provenance should be grouped. That in turn is non-trivial due to the way data elements will be built and updated:

- Any given data element will be the result of the processing of raw data by a Task Graph (also called a pipeline). Each Graph consists of many Tasks. Each task may involve executing multiple algorithms.
- It is anticipated we will need to fix bad data (as long as it is unreleased) by reprocessing it (e.g. using a fixed algorithm), so in practice different parts of the same data element might have many different processing histories.
- Finally, there is a large number of dependencies. Here is just a small example: object data depends on a coadd, which is generated using masks derived from difference sources which depend on the corresponding difference image which depends on a template image which depends on calibrated science images and so on...

Also, provenance should be flexible enough to easily allow adding new items to the system state.

For the above reasons, simple solutions like timestamping every data element will not work.

In practice, we plan to implement the state recording function of Provenance using time ranges, with granularity up to 1 sec. Every configurable piece of information will be tracked through two tables: one that keeps definitions and the other that keeps configurations. For example, each filter (u, g, r, i, z, y) will be represented as one row in `prv_Filter` table (there will be exactly 6 definitions), and each configuration of each filter will be represented as one row in `prv_cnf_Filter` table (initially there will be 6 configurations: one for each filter, but if we happen to later break one filter and will introduce a new one, a new configuration will need to be inserted). Each configuration will have a validity time range assigned (time period when it was valid).

The heart of the processing recording function of Provenance is the *prv_ProcHistory* table; it has two responsibilities:

- Assign a unique processing history id (*procHistoryId*) for each creation of a set of output data elements. An 8-byte integer is sufficient to provide the needed granularity.
- Bind each unique *procHistoryId* with a set of executions of tasks, including the inputs to those tasks, identified by their own *procHistoryIds*. (A task is the smallest chunk that will be executed atomically, that is, if part of a task fails, the whole task will be rolled back and re-run). Typically there will be one task execution for each *procHistoryId*, but in cases of reprocessing there may be more than one.

Notice that *procHistoryId* is not associated with a time range, but with a set of task executions, each having its own configuration, its own time range, and its own node it runs on.

The Provenance may also track which columns for each table are updated by each stage (we require that each column is updated by a single stage only - for that reason some columns, eg. flags had to be split into multiple columns).

Every data element that needs to have provenance tracked (for example, every row in the Object, Source, or Exposure table) will need to store a *procHistoryId*. Note that a single row might be generated or updated by different pipelines or task graphs, and each may run at different time, on different hardware, possibly even at a different site. Having a *procHistoryId* will allow it to find which task graphs and tasks were executed for it, at what time, and on which processing nodes. It will then be possible to correlate the time periods when each task was executed with the configurations that were valid at those times.

Such an approach has two important pros:

- **Flexibility:** It allows us to decouple configurable elements from Objects, Sources, Exposures, etc. — they are very loosely coupled through time range only. In particular, it means that new configurable items can be added or removed from the system at any given time with no need to do schema evolution and no need to update existing data.
- **Space-efficiency:** It is clear that components of the system will be changing relatively frequently. Minimizing the storage required to track these changes is essential. In our

case, a new configuration will only require storage for that configuration. Each Object/Source/Exposure only needs to store one procHistoryId to get access to hundreds of different configurations.

We expect we will need to tune the current approach to provide efficient query access to the Provenance. In particular, the implementation relies on range queries which may need special optimizations, or special indexes.

4.2 Schema Evolution

In summary, we expect to avoid heavy changes that alter shape of large tables (e.g. by pre-allocating space), and rely on partitioning to scale and run necessary schema changes fast and scalably in parallel⁴.

4.2.1 Released Data Sets

Each data release is independent, therefore the database schema can easily change between data releases; we do not anticipate any non-trivial challenges here.

Adding new columns. We are planning to keep a few extra unused columns of each common type (for example, 3 FLOATs, 3 DOUBLEs, 3 INTs, 3 BIGINTs) for each large table, and use one of them when a new column is needed. (renaming a column is a trivial and instantaneous operation). Speed of filling new columns should be comparable to speed of a full table scan.

Updating existing columns. Speed of updating values should be comparable to speed of a single shared scan.

Deleting columns. If we have to delete a column, instead of deleting it, (which is expensive as it changes shape of a table), we will add it to the pool of "extra, unused columns" by renaming it.

The unused/hidden columns will be hidden through non-materialized views. It is likely that we will end up providing multiple views, e.g. each time we make a schema change, we'd expose

⁴For more information about the schema evolution tests we run, refer to <https://dev.lsstcorp.org/trac/wiki/db/tests/SchemaEvolution>

the changes through a new view.

Non-trivial changes. Should we ever need to do non-trivial changes: Non trivial schema changes on multi-billion row table tend to take extremely long time (measured in days). In our case, each large table will be partitioned into several thousand chunks, which makes the updates scalable and faster: not only the updates can be done in parallel on multiple machines, but each partition is small enough to be handled efficiently without running into any bottlenecks or problems with lack of buffer space.

Notes on administration. All queries accessing the database will go through our interface, which intercepts and interprets the queries. This will allow us to put the entire system in special mode (user queries will be queued, administrative queries will continue to work). The administrative tools we are building will allow us to push the necessary updates to all nodes, and we could use the existing scheduling mechanisms to run the upgrade in the most efficient fashion. The admin tools will also allow running necessary verifications (e.g., whether the schema for all table chunks matches).

4.2.2 Level 1 (Up-to-date Catalog)

We will have opportunity to update schema every time we are replacing Level 1 catalog after each data release. Should we need to make the updates in between, the changes can easily be done during the time when we are not taking data (daytime).

5 Naming Conventions

We have defined a set of naming convention for naming tables, columns and relationships. In summary, the format of table names follows:

<_><prefix><purpose>_<extend>_<component><postfix>

Leading _ is used to show it is an internal table.

Prefix:

- All Provenance tables are prefixed with `prv_`, example: `prv_Amp`. Further, it may be followed by `cnf_` which is used for provenance tables containing individual configurations. For example, one row in `prv_Ccd` contains information about one particular Ccd, and `prv_cnf_Ccd` contains information about one specific configuration version of one particular Ccd.

Purpose: it includes things like Raw, Science, Calibration, Bias, Dark, Flat, Fringe, IR. (we have recently dropped "Science" for *Visit tables).

Extend: Fpa, Ccd, Amp (we have dropped "Fpa" from raw and science table names).

Component: Exposure, Object, Source, Psf, Wcs...

Postfix: optional

Mapping tables follows its own naming scheme: `_<table X name>_To_<table Y name>`

6 Conventions for NULL and missing data

In addition to defining the table columns, an important component of the data model is standardizing the representation of missing and out-of-range values. These conventions must cover the variety of formats used for storing the data during processing (prior to loading into a database), as well as any other formats that data may be presented to the users. In practice, this means we must pay close attention to how data are stored in FITS files (during processing) and Parquet files (from which the relational database is loaded, and which may also be available to users). Loading tables into a database may also involve creating intermediate CSV files; these are not user-facing, but we include requirements on CSV files here to reduce the potential for discrepancies in NULL handling.

In the following conventions, "nullable" is used as a synonym for **not** applying a NOT NULL constraint to a column.

1. Float columns must use NaN in FITS and Parquet to represent unknown or invalid values, which must be converted to NULL in an RDBMS. RDBMS loading must also accept +/-

Infinity and convert them to NULL.

Rationale: For floating point columns, both IEEE 754 and RDBMS's can represent invalid values without resorting to sentinel values.

2. Integer columns must only be nullable if they are foreign keys to other tables. Integer measurements that need to report an error condition should use a flag column. Ints cannot be NaN.

Rationale: There is no standard for representing invalid data as integers, other than assigning a sentinel. RDBMS's can report NULL, but this has no equivalent in FITS or parquet. Most uses of ints in measurements are quantities like a count of the pixels in a source, which can be meaningfully zero when e.g. no data is available.

3. Foreign key columns must use a value of zero to represent "no linked row" in FITS and Parquet, which must be converted to NULL in an RDBMS.

Rationale: A NULL in the RDBMS is required to prevent spurious joins.

4. Parquet integers must always be signed, to support Java readers.

Rationale: Some python libraries are capable of writing unsigned integers to parquet files, but this is highly non-portable.

5. CSV files must use the string "\N" to represent NULL. CSV files must not contain +/-Infinity or NaN.

Rationale: Many other strings would work, but adopting a single convention will minimize conflicts.

6. +/-Infinity must not be meaningfully distinct from NaN in pipeline outputs.

Rationale: The SQL standard only has NULL, and cannot distinguish between +/-Infinity and NaN.

7. RDBMS schemas must assume that all FLOAT pipeline columns are nullable.

Rationale: any enforcement of required values should be performed at the pipeline processing stage. If undesired NULLs are present in measurement outputs, then catching them at database-load time is too late.

7 Baseline Schema

This schema corresponds to git commit 82029d8 in the sdm_schemas package.

7.1 ZZZ_Schema_Description

Internal table used for storing schema version and description.

Table 1: ZZZ_Schema_Description Table

Column Name	Data type	Unit	Description [UCD]
version	string		The schema version.
descr	string		Description of the schema.

7.2 DiaObject

The DiaObject table contains descriptions of the astronomical objects detected on one or more difference images.

Table 2: DiaObject Table

Column Name	Data type	Unit	Description [UCD]
diaObjectId	long		Unique id. [meta.id; src]
validityStart	timestamp		Time when validity of this diaObject starts.
validityEnd	timestamp		Time when validity of this diaObject ends.
ra	double	deg	RA-coordinate of the position of the object at time radecTai. [pos.eq.ra]
raSigma	float	deg	Uncertainty of ra. [stat.error; pos.eq.ra]
decl	double	deg	Decl-coordinate of the position of the object at time radecTai. [pos.eq.dec]
declSigma	float	deg	Uncertainty of decl. [stat.error; pos.eq.dec]
ra_decl_Cov	float	deg ²	Covariance between ra and decl.
radecTai	double		Time at which the object was at a position ra/decl. [time.epoch]
pmRa	float	mas/yr	Proper motion (ra). [pos.pm]
pmRaSigma	float	mas/yr	Uncertainty of pmRa. [stat.error; pos.pm]
pmDecl	float	mas/yr	Proper motion (decl). [pos.pm]
pmDeclSigma	float	mas/yr	Uncertainty of pmDecl. [stat.error; pos.pm]
parallax	float	mas	Parallax. [pos.parallax]

Continued on next page

Table 2: DiaObject Table

Column Name	Data type	Unit	Description [UCD]
parallaxSigma	float	mas	Uncertainty of parallax. [stat.error; pos.parallax]
pmRa_pmDecl_Cov	float	(mas/yr) ²	Covariance of pmRa and pmDecl. [stat.covariance; pos.eq]
pmRa_parallax_Cov	float	mas ² /yr	Covariance of pmRa and parallax. [stat.covariance]
pmDecl_parallax_Cov	float	mas ² /yr	Covariance of pmDecl and parallax. [stat.covariance]
pmParallaxLnL	float		Natural log of the likelihood of the linear proper motion parallax fit. [stat.likelihood]
pmParallaxChi2	float		Chi ² static of the model fit. [stat.fit.chi2]
pmParallaxNdata	int		The number of data points used to fit the model.
uPSFluxMean	float	nmgy	Weighted mean point-source model magnitude for u filter. [phot.count]
uPSFluxMeanErr	float	nmgy	Standard error of uPSFluxMean. [stat.error]
uPSFluxSigma	float	nmgy	Standard deviation of the distribution of uPSFlux. [stat.stdev]
uPSFluxChi2	float		Chi ² statistic for the scatter of uPSFlux around uPSFluxMean. [stat.fit.chi2]
uPSFluxNdata	int		The number of data points used to compute uPSFluxChi2.
uFPFluxMean	float	nmgy	Weighted mean forced photometry flux for u filter. [phot.count]
uFPFluxMeanErr	float	nmgy	Standard error of uFPFluxMean. [stat.error]
uFPFluxSigma	float	nmgy	Standard deviation of the distribution of uFPFlux. [stat.stdev]
gPSFluxMean	float	nmgy	Weighted mean point-source model magnitude for g filter. [phot.count]
gPSFluxMeanErr	float	nmgy	Standard error of gPSFluxMean. [stat.error]
gPSFluxSigma	float	nmgy	Standard deviation of the distribution of gPSFlux. [stat.stdev]

Continued on next page

Table 2: DiaObject Table

Column Name	Data type	Unit	Description [UCD]
gPSFluxChi2	float		Chi ² statistic for the scatter of gPSFlux around gPSFluxMean. [stat.fit.chi2]
gPSFluxNdata	int		The number of data points used to compute gPSFluxChi2.
gFPFluxMean	float	nmgy	Weighted mean forced photometry flux for g filter. [phot.count]
gFPFluxMeanErr	float	nmgy	Standard error of gFPFluxMean. [stat.error]
gFPFluxSigma	float	nmgy	Standard deviation of the distribution of gFPFlux. [stat.stdev]
rPSFluxMean	float	nmgy	Weighted mean point-source model magnitude for r filter. [phot.count]
rPSFluxMeanErr	float	nmgy	Standard error of rPSFluxMean. [stat.error]
rPSFluxSigma	float	nmgy	Standard deviation of the distribution of rPSFlux. [stat.stdev]
rPSFluxChi2	float		Chi ² statistic for the scatter of rPSFlux around rPSFluxMean. [stat.fit.chi2]
rPSFluxNdata	int		The number of data points used to compute rPSFluxChi2.
rFPFluxMean	float	nmgy	Weighted mean forced photometry flux for r filter. [phot.count]
rFPFluxMeanErr	float	nmgy	Standard error of rFPFluxMean. [stat.error]
rFPFluxSigma	float	nmgy	Standard deviation of the distribution of rFPFlux. [stat.stdev]
iPSFluxMean	float	nmgy	Weighted mean point-source model magnitude for i filter. [phot.count]
iPSFluxMeanErr	float	nmgy	Standard error of iPSFluxMean. [stat.error]
iPSFluxSigma	float	nmgy	Standard deviation of the distribution of iPSFlux. [stat.stdev]
iPSFluxChi2	float		Chi ² statistic for the scatter of iPSFlux around iPSFluxMean. [stat.fit.chi2]
iPSFluxNdata	int		The number of data points used to compute iPSFluxChi2.

Continued on next page

Table 2: DiaObject Table

Column Name	Data type	Unit	Description [UCD]
iFPFluxMean	float	nmgy	Weighted mean forced photometry flux for i filter. [phot.count]
iFPFluxMeanErr	float	nmgy	Standard error of iFPFluxMean. [stat.error]
iFPFluxSigma	float	nmgy	Standard deviation of the distribution of iFPFlux. [stat.stdev]
zPSFluxMean	float	nmgy	Weighted mean point-source model magnitude for z filter. [phot.count]
zPSFluxMeanErr	float	nmgy	Standard error of zPSFluxMean. [stat.error]
zPSFluxSigma	float	nmgy	Standard deviation of the distribution of zPSFlux. [stat.stdev]
zPSFluxChi2	float		Chi ² statistic for the scatter of zPSFlux around zPSFluxMean. [stat.fit.chi2]
zPSFluxNdata	int		The number of data points used to compute zPSFluxChi2.
zFPFluxMean	float	nmgy	Weighted mean forced photometry flux for z filter. [phot.count]
zFPFluxMeanErr	float	nmgy	Standard error of zFPFluxMean. [stat.error]
zFPFluxSigma	float	nmgy	Standard deviation of the distribution of zFPFlux. [stat.stdev]
yPSFluxMean	float	nmgy	Weighted mean point-source model magnitude for y filter. [phot.count]
yPSFluxMeanErr	float	nmgy	Standard error of yPSFluxMean. [stat.error]
yPSFluxSigma	float	nmgy	Standard deviation of the distribution of yPSFlux. [stat.error]
yPSFluxChi2	float		Chi ² statistic for the scatter of yPSFlux around yPSFluxMean. [stat.fit.chi2]
yPSFluxNdata	int		The number of data points used to compute yPSFluxChi2.
yFPFluxMean	float	nmgy	Weighted mean forced photometry flux for y filter. [phot.count]
yFPFluxMeanErr	float	nmgy	Standard error of yFPFluxMean. [stat.error]

Continued on next page

Table 2: DiaObject Table

Column Name	Data type	Unit	Description [UCD]
yFPFluxSigma	float	nmgy	Standard deviation of the distribution of yFPFlux. [stat.stdev]
uLcPeriodic	binary		Periodic features extracted from light-curves using generalized Lomb-Scargle periodogram for u filter. [32 FLOAT].
gLcPeriodic	binary		Periodic features extracted from light-curves using generalized Lomb-Scargle periodogram for g filter. [32 FLOAT].
rLcPeriodic	binary		Periodic features extracted from light-curves using generalized Lomb-Scargle periodogram for r filter. [32 FLOAT].
iLcPeriodic	binary		Periodic features extracted from light-curves using generalized Lomb-Scargle periodogram for i filter. [32 FLOAT].
zLcPeriodic	binary		Periodic features extracted from light-curves using generalized Lomb-Scargle periodogram for z filter. [32 FLOAT].
yLcPeriodic	binary		Periodic features extracted from light-curves using generalized Lomb-Scargle periodogram for y filter. [32 FLOAT].
uLcNonPeriodic	binary		Non-periodic features extracted from light-curves using generalized Lomb-Scargle periodogram for u filter. [20 FLOAT].
gLcNonPeriodic	binary		Non-periodic features extracted from light-curves using generalized Lomb-Scargle periodogram for g filter. [20 FLOAT].
rLcNonPeriodic	binary		Non-periodic features extracted from light-curves using generalized Lomb-Scargle periodogram for r filter. [20 FLOAT].
iLcNonPeriodic	binary		Non-periodic features extracted from light-curves using generalized Lomb-Scargle periodogram for i filter. [20 FLOAT].

Continued on next page

Table 2: DiaObject Table

Column Name	Data type	Unit	Description [UCD]
zLcNonPeriodic	binary		Non-periodic features extracted from light-curves using generalized Lomb-Scargle periodogram for z filter. [20 FLOAT].
yLcNonPeriodic	binary		Non-periodic features extracted from light-curves using generalized Lomb-Scargle periodogram for y filter. [20 FLOAT].
nearbyObj1	long		Id of the closest nearby object. [meta.id; src]
nearbyObj1Dist	float	arcsec	Distance to nearbyObj1.
nearbyObj1LnP	float		Natural log of the probability that the observed diaObject is the same as the nearbyObj1.
nearbyObj2	long		Id of the second-closest nearby object. [meta.id; src]
nearbyObj2Dist	float	arcsec	Distance to nearbyObj2.
nearbyObj2LnP	float		Natural log of the probability that the observed diaObject is the same as the nearbyObj2.
nearbyObj3	long		Id of the third-closest nearby object. [meta.id; src]
nearbyObj3Dist	float	arcsec	Distance to nearbyObj3.
nearbyObj3LnP	float		Natural log of the probability that the observed diaObject is the same as the nearbyObj3.
flags	long		Flags, bitwise OR tbd. [meta.code]
htmlId20	long		HTM index.

7.3 SObject

The SObject table contains description of the Solar System (moving) Objects.

Table 3: SObject Table

Column Name	Data type	Unit	Description [UCD]
ssObjectId	long		Unique identifier. [meta.id; src]
q	double		Osculating orbital elements at epoch (q, e, i, lan, aop, M, epoch).

Continued on next page

Table 3: SObject Table

Column Name	Data type	Unit	Description [UCD]
qSigma	double		Uncertainty of q. [stat.error]
e	double		Osculating orbital elements at epoch (q, e, i, lan, aop, M, epoch).
eSigma	double		Uncertainty of e. [stat.error]
i	double		Osculating orbital elements at epoch (q, e, i, lan, aop, M, epoch).
iSigma	double		Uncertainty of i. [stat.error]
lan	double		Osculating orbital elements at epoch (q, e, i, lan, aop, M, epoch).
lanSigma	double		Uncertainty of lan. [stat.error]
aop	double		Osculating orbital elements at epoch (q, e, i, lan, aop, M, epoch).
aopSigma	double		Uncertainty of aop. [stat.error]
M	double		Osculating orbital elements at epoch (q, e, i, lan, aop, M, epoch).
MSigma	double		Uncertainty of M. [stat.error]
epoch	double		Osculating orbital elements at epoch (q, e, i, lan, aop, M, epoch).
epochSigma	double		Uncertainty of epoch. [stat.error]
q_e_Cov	double		Covariance of q and e. [stat.covariance]
q_i_Cov	double		Covariance of q and i. [stat.covariance]
q_lan_Cov	double		Covariance of q and lan. [stat.covariance]
q_aop_Cov	double		Covariance of q and aop. [stat.covariance]
q_M_Cov	double		Covariance of q and M. [stat.covariance]
q_epoch_Cov	double		Covariance of q and epoch. [stat.covariance]
e_i_Cov	double		Covariance of e and i. [stat.covariance]
e_lan_Cov	double		Covariance of e and lan. [stat.covariance]
e_aop_Cov	double		Covariance of e and aop. [stat.covariance]
e_M_Cov	double		Covariance of e and M. [stat.covariance]
e_epoch_Cov	double		Covariance of e and epoch. [stat.covariance]
i_lan_Cov	double		Covariance of i and lan. [stat.covariance]
i_aop_Cov	double		Covariance of i and aop. [stat.covariance]

Continued on next page

Table 3: SObject Table

Column Name	Data type	Unit	Description [UCD]
i_M_Cov	double		Covariance of i and M. [stat.covariance]
i_epoch_Cov	double		Covariance of i and epoch. [stat.covariance]
lan_aop_Cov	double		Covariance of lan and aop. [stat.covariance]
lan_M_Cov	double		Covariance of lan and M. [stat.covariance]
lan_epoch_Cov	double		Covariance of lan and epoch. [stat.covariance]
aop_M_Cov	double		Covariance of aop and M. [stat.covariance]
aop_epoch_Cov	double		Covariance of aop and epoch. [stat.covariance]
M_epoch_Cov	double		Covariance of M and epoch. [stat.covariance]
arc	float	days	Arc of observation.
orbFitLnL	float		Natural log of the likelihood of the orbital elements fit. [stat.likelihood]
orbFitChi2	float		Chi ² statistic of the orbital elements fit. [stat.fit.chi2]
orbFitNdata	int		Number of observations used in the fit.
MOID1	float	AU	Minimum orbit intersection distance.
MOID2	float	AU	Minimum orbit intersection distance.
moidLon1	double	deg	MOID longitudes.
moidLon2	double	deg	MOID longitudes.
uH	float	mag	Mean absolute magnitude for u filter.
uHErr	float	mag	Uncertainty of uH estimate. [stat.error]
uG1	float	mag	Fitted G1 slope parameter for u filter.
uG1Err	float	mag	Uncertainty of uG1 estimate. [stat.error]
uG2	float	mag	Fitted G2 slope parameter for u filter.
uG2Err	float	mag	Uncertainty of uG2 estimate. [stat.error]
gH	float	mag	Mean absolute magnitude for g filter.
gHErr	float	mag	Uncertainty of gH estimate. [stat.error]
gG1	float	mag	Fitted G1 slope parameter for g filter.
gG1Err	float	mag	Uncertainty of gG1 estimate. [stat.error]
gG2	float	mag	Fitted G2 slope parameter for g filter.
gG2Err	float	mag	Uncertainty of gG2 estimate. [stat.error]
rH	float	mag	Mean absolute magnitude for r filter.
rHErr	float	mag	Uncertainty of rH estimate. [stat.error]

Continued on next page

Table 3: SObject Table

Column Name	Data type	Unit	Description [UCD]
rG1	float	mag	Fitted G1 slope parameter for r filter.
rG1Err	float	mag	Uncertainty of rG1 estimate. [stat.error]
rG2	float	mag	Fitted G2 slope parameter for r filter.
rG2Err	float	mag	Uncertainty of rG2 estimate. [stat.error]
iH	float	mag	Mean absolute magnitude for i filter.
iHErr	float	mag	Uncertainty of iH estimate. [stat.error]
iG1	float	mag	Fitted G1 slope parameter for i filter.
iG1Err	float	mag	Uncertainty of iG1 estimate. [stat.error]
iG2	float	mag	Fitted G2 slope parameter for i filter.
iG2Err	float	mag	Uncertainty of iG2 estimate. [stat.error]
zH	float	mag	Mean absolute magnitude for z filter.
zHErr	float	mag	Uncertainty of zH estimate. [stat.error]
zG1	float	mag	Fitted G1 slope parameter for z filter.
zG1Err	float	mag	Uncertainty of zG1 estimate. [stat.error]
zG2	float	mag	Fitted G2 slope parameter for z filter.
zG2Err	float	mag	Uncertainty of zG2 estimate. [stat.error]
yH	float	mag	Mean absolute magnitude for y filter.
yHErr	float	mag	Uncertainty of yH estimate. [stat.error]
yG1	float	mag	Fitted G1 slope parameter for y filter.
yG1Err	float	mag	Uncertainty of yG1 estimate. [stat.error]
yG2	float	mag	Fitted G2 slope parameter for y filter.
yG2Err	float	mag	Uncertainty of yG2 estimate. [stat.error]
flags	long		Flags, bitwise OR tbd. [meta.code]

7.4 DiaSource

Table to store 'difference image sources'; - sources detected at SNR ≥ 5 on difference images.

Table 4: DiaSource Table

Column Name	Data type	Unit	Description [UCD]
diaSourceId	long		Unique id. [meta.id; obs.image]

Continued on next page

Table 4: DiaSource Table

Column Name	Data type	Unit	Description [UCD]
ccdVisitId	long		Id of the ccdVisit where this diaSource was measured. Note that we are allowing a diaSource to belong to multiple amplifiers, but it may not span multiple ccds. [meta.id; obs.image]
diaObjectId	long		Id of the diaObject this source was associated with, if any. If not, it is set to NULL (each diaSource will be associated with either a diaObject or ssObject). [meta.id; src]
ssObjectId	long		Id of the ssObject this source was associated with, if any. If not, it is set to NULL (each diaSource will be associated with either a diaObject or ssObject). [meta.id; src]
parentDiaSourceId	long		Id of the parent diaSource this diaSource has been deblended from, if any. [meta.id; src]
prv_procOrder	int		Position of this diaSource in the processing order relative to other diaSources within a given diaObjectId or ssObjectId.
ssObjectReassocTime	timestamp		Time when this diaSource was reassociated from diaObject to ssObject (if such reassociation happens, otherwise NULL).
midPointTai	double	d	Effective mid-exposure time for this diaSource. [time.epoch]
ra	double	deg	RA-coordinate of the center of this diaSource. [pos.eq.ra]
raSigma	float	deg	Uncertainty of ra. [stat.error; pos.eq.ra]
decl	double	deg	Decl-coordinate of the center of this diaSource. [pos.eq.dec]
declSigma	float	deg	Uncertainty of decl. [stat.error; pos.eq.dec]
ra_decl_Cov	float	deg ²	Covariance between ra and decl. [stat.covariance]
x	float	pixel	x position computed by a centroiding algorithm. [pos.cartesian.x]

Continued on next page

Table 4: DiaSource Table

Column Name	Data type	Unit	Description [UCD]
xSigma	float	pixel	Uncertainty of x. [stat.error:pos.cartesian.x]
y	float	pixel	y position computed by a centroiding algorithm. [pos.cartesian.y]
ySigma	float	pixel	Uncertainty of y. [stat.error:pos.cartesian.y]
x_y_Cov	float	pixel ²	Covariance between x and y. [stat.covariance]
apFlux	float	nmgy	Calibrated aperture flux. Note that this actually measures the difference between the template and the visit image. [phot.count]
apFluxErr	float	nmgy	Estimated uncertainty of apFlux. [stat.error; phot.count]
snr	float		The signal-to-noise ratio at which this source was detected in the difference image. [stat.snr]
psFlux	float	nmgy	Calibrated flux for Point Source model. Note this actually measures the flux difference between the template and the visit image. [phot.count]
psFluxSigma	float	nmgy	Uncertainty of psFlux.
psRa	double	deg	RA-coordinate of centroid for point source model. [pos.eq.ra]
psRaSigma	float	deg	Uncertainty of psRa. [stat.error; pos.eq.ra]
psDecl	double	deg	Decl-coordinate of centroid for point source model. [pos.eq.dec]
psDeclSigma	float	deg	Uncertainty of psDecl. [stat.error; pos.eq.dec]
psFlux_psRa_Cov	float	ngmy*deg	Covariance between psFlux and psRa. [stat.covariance]
psFlux_psDecl_Cov	float	ngmy*deg	Covariance between psFlux and psDecl. [stat.covariance]
psRa_psDecl_Cov	float	deg ²	Covariance between psRa and psDecl. [stat.covariance]
psLnL	float		Natural log likelihood of the observed data given the Point Source model. [stat.likelihood]
psChi2	float		Chi ² statistic of the model fit. [stat.fit.chi2]

Continued on next page

Table 4: DiaSource Table

Column Name	Data type	Unit	Description [UCD]
psNdata	int		The number of data points (pixels) used to fit the model.
trailFlux	float	nmgy	Calibrated flux for a trailed source model. Note this actually measures the flux difference between the template and the visit image.
trailFluxSigma	float	nmgy	Uncertainty of trailFlux.
trailRa	double	deg	RA-coordinate of centroid for trailed source model. [pos.eq.ra]
trailRaSigma	float	deg	Uncertainty of trailRa. [stat.error; pos.eq.ra]
trailDecl	double	deg	Decl-coordinate of centroid for trailed source model. [pos.eq.dec]
trailDeclSigma	float	deg	Uncertainty of trailDecl. [stat.error; pos.eq.dec]
trailLength	float	arcsec	Maximum likelihood fit of trail length.
trailLengthSigma	float	nmgy	Uncertainty of trailLength.
trailAngle	float	deg	Maximum likelihood fit of the angle between the meridian through the centroid and the trail direction (bearing).
trailAngleSigma	float	nmgy	Uncertainty of trailAngle.
trailFlux_trailRa_Cov	float		Covariance of trailFlux and trailRa. [stat.covariance]
trailFlux_trailDecl_Cov	float		Covariance of trailFlux and trailDecl. [stat.covariance]
trailFlux_trailLength_Cov	float		Covariance of trailFlux and trailLength [stat.covariance]
trailFlux_trailAngle_Cov	float		Covariance of trailFlux and trailAngle [stat.covariance]
trailRa_trailDecl_Cov			

Continued on next page

Table 4: DiaSource Table

Column Name	Data type	Unit	Description [UCD]
trailRa_trailDecl_Cov	float		Covariance of trailRa and trailDecl. [stat.covariance]
trailRa_trailLength_Cov	float		Covariance of trailRa and trailLength. [stat.covariance]
trailRa_trailAngle_Cov	float		Covariance of trailRa and trailAngle. [stat.covariance]
trailDecl_trailLength_Cov	float		Covariance of trailDecl and trailLength. [stat.covariance]
trailDecl_trailAngle_Cov	float		Covariance of trailDecl and trailAngle. [stat.covariance]
trailLength_trailAngle_Cov	float		Covariance of trailLength and trailAngle [stat.covariance]
trailLnL	float		Natural log likelihood of the observed data given the trailed source model. [stat.likelihood]
trailChi2	float		Chi ² statistic of the model fit. [stat.fit.chi2]
trailNdata	int		The number of data points (pixels) used to fit the model.
dipMeanFlux	float	nmgy	Maximum likelihood value for the mean absolute flux of the two lobes for a dipole model.
dipMeanFluxSigma	float	nmgy	Uncertainty of dipMeanFlux. [stat.error]
dipFluxDiff	float	nmgy	Maximum likelihood value for the difference of absolute fluxes of the two lobes for a dipole model.
dipFluxDiffSigma	float	nmgy	Uncertainty of dipFluxDiff. [stat.error]
dipRa	double	deg	RA-coordinate of centroid for dipole model. [pos.eq.ra]
dipRaSigma	float	deg	Uncertainty of trailRa. [stat.error; pos.eq.ra]

Continued on next page

Table 4: DiaSource Table

Column Name	Data type	Unit	Description [UCD]
dipDecl	double	deg	Decl-coordinate of centroid for dipole model. [pos.eq.dec]
dipDeclSigma	float	deg	Uncertainty of dipDecl. [stat.error; pos.eq.dec]
dipLength	float	arcsec	Maximum likelihood value for the lobe separation in dipole model. [pos.angDistance]
dipLengthSigma	float	arcsec	Uncertainty of dipLength. [stat.error; pos.angDistance]
dipAngle	float	deg	Maximum likelihood fit of the angle between the meridian through the centroid and the dipole direction (bearing, from negative to positive lobe). [pos.posAng]
dipAngleSigma	float	deg	Uncertainty of dipAngle. [stat.error; pos.posAng]
dipMeanFlux_dipFluxDiff_Cov	float		Covariance of dipMeanFlux and dipFluxDiff. [stat.covariance]
dipMeanFlux_dipRa_Cov	float		Covariance of dipMeanFlux and dipRa. [stat.covariance]
dipMeanFlux_dipDecl_Cov	float		Covariance of dipMeanFlux and dipDecl. [stat.covariance]
dipMeanFlux_dipLength_Cov	float		Covariance of dipMeanFlux and dipLength. [stat.covariance]
dipMeanFlux_dipAngle_Cov	float		Covariance of dipMeanFlux and dipAngle. [stat.covariance]
dipFluxDiff_dipRa_Cov	float		Covariance of dipFluxDiff and dipRa. [stat.covariance]
dipFluxDiff_dipDecl_Cov			

Continued on next page

Table 4: DiaSource Table

Column Name	Data type	Unit	Description [UCD]
	float		Covariance of dipFluxDiff and dipDecl. [stat.covariance]
dipFluxDiff_dipLength_Cov	float		Covariance of dipFluxDiff and dipLength. [stat.covariance]
dipFluxDiff_dipAngle_Cov	float		Covariance of dipFluxDiff and dipAngle. [stat.covariance]
dipRa_dipDecl_Cov	float		Covariance of dipRa and dipDecl. [stat.covariance]
dipRa_dipLength_Cov	float		Covariance of dipRa and dipLength. [stat.covariance]
dipRa_dipAngle_Cov	float		Covariance of dipRa and dipAngle. [stat.covariance]
dipDecl_dipLength_Cov	float		Covariance of dipDecl and dipLength. [stat.covariance]
dipDecl_dipAngle_Cov	float		Covariance of dipDecl and dipAngle. [stat.covariance]
dipLength_dipAngle_Cov	float		Covariance of dipLength and dipAngle. [stat.covariance]
dipLnL	float		Natural log likelihood of the observed data given the dipole source model. [stat.likelihood]
dipChi2	float		Chi ² statistic of the model fit. [stat.fit.chi2]
dipNdata	int		The number of data points (pixels) used to fit the model.
totFlux	float	nmg	Calibrated flux for Point Source model measured on the visit image centered at the centroid measured on the difference image (forced photometry flux). [phot.count]

Continued on next page

Table 4: DiaSource Table

Column Name	Data type	Unit	Description [UCD]
totFluxErr	float	nmgy	Estimated uncertainty of totFlux. [stat.error; phot.count]
diffFlux	float	nmgy	Calibrated flux for Point Source model centered on radec but measured on the difference of snaps comprising this visit.
diffFluxErr	float	nmgy	Estimated uncertainty of diffFlux. [stat.error; phot.count]
fpBkgd	float	nmgy/asec ²	Estimated sky background at the position (centroid) of the object.
fpBkgdErr	float	nmgy/asec ²	Estimated uncertainty of fpBkgd.
ixx	float	nmgy*asec ²	Adaptive second moment of the source intensity.
ixxSigma	float	nmgy*asec ²	Uncertainty of ixx. [stat.error]
iyx	float	nmgy*asec ²	Adaptive second moment of the source intensity.
iyxSigma	float	nmgy*asec ²	Uncertainty of iyx. [stat.error]
ixy	float	nmgy*asec ²	Adaptive second moment of the source intensity.
ixySigma	float	nmgy*asec ²	Uncertainty of ixy. [stat.error]
ixx_iyy_Cov	float	nmgy ² *asec ⁴	Covariance of ixx and iyy.
ixx_ixy_Cov	float	nmgy ² *asec ⁴	Covariance of ixx and ixy.
iyx_ixy_Cov	float	nmgy ² *asec ⁴	Covariance of iyx and ixy.
ixxPSF	float	nmgy*asec ²	Adaptive second moment for the PSF.
iyxPSF	float	nmgy*asec ²	Adaptive second moment for the PSF.
ixyPSF	float	nmgy*asec ²	Adaptive second moment for the PSF.
extendedness	float		A measure of extendedness, Computed using a combination of available moments and model fluxes or from a likelihood ratio of point/trailed source models (exact algorithm TBD). extendedness = 1 implies a high degree of confidence that the source is extended. extendedness = 0 implies a high degree of confidence that the source is point-like.

Continued on next page

Table 4: DiaSource Table

Column Name	Data type	Unit	Description [UCD]
spuriousness	float		A measure of spuriousness, computed using information from the source and image characterization, as well as the information on the Telescope and Camera system (e.g., ghost maps, defect maps, etc.).
flags	long		Flags, bitwise OR tbd. [meta.code]
htmlId20	long		HTM index.

7.5 DiaForcedSource

Forced-photometry source measurement on an individual difference Exposure based on a Multifit shape model derived from a deep detection.

Table 5: DiaForcedSource Table

Column Name	Data type	Unit	Description [UCD]
diaObjectId	long		[meta.id; src]
ccdVisitId	long		Id of the visit where this forcedSource was measured. Note that we are allowing a forcedSource to belong to multiple amplifiers, but it may not span multiple ccDs. [meta.id; obs.image]
psFlux	float	nmgy	Point Source model flux. [phot.count]
psFluxSigma	float	nmgy	Uncertainty of psFlux. [stat.error; phot.count]
x	float	pixel	x position at which psFlux has been measured. [pos.cartesian.x]
y	float	pixel	y position at which psFlux has been measured. [pos.cartesian.y]
flags	long		Flags, bitwise OR tbd [meta.code]

7.6 DiaObject_To_Object_Match

The table stores mapping of diaObjects to the nearby objects.

Table 6: DiaObject_To_Object_Match Table

Column Name	Data type	Unit	Description [UCD]
diaObjectId	long		Id of diaObject. [meta.id; src]
objectId	long		Id of a nearby object. [meta.id; src]
dist	float	arcsec	The distance between the diaObject and the object.
lnP	float		Natural log of the probability that the observed diaObject is the same as the nearby object.

7.7 Object

The Object table contains descriptions of the multi-epoch static astronomical objects, in particular their astrophysical properties as derived from analysis of the Sources that are associated with them. Note that fast moving objects are kept in the MovingObject tables. Note that less-frequently used columns are stored in a separate table called Object_Extra.

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
objectId	long		Unique id. [meta.id; src]
parentObjectId	long		Id of the parent object this object has been deblended from, if any.
prv_inputId	int		Pointer to prv_InputType. Indicates which input was used to produce a given object.
psRa	double	deg	RA-coordinate of the center of the object for the Point Source model at time 'psEpoch'. [pos.eq.ra]
psRaSigma	float	deg	Uncertainty of psRa. [stat.error; pos.eq.ra]
psDecl	double	deg	Decl-coordinate of the center of the object for the Point Source model at time 'psEpoch'. [pos.eq.dec]
psDeclSigma	float	deg	Uncertainty of psDecl. [stat.error; pos.eq.dec]

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
psMuRa	float	mas/yr	Proper motion (ra) for the Point Source model. [pos.pm]
psMuRaSigma	float	mas/yr	Uncertainty of psMuRa. [stat.error; pos.pm]
psMuDecl	float	mas/yr	Proper motion (decl) for the Point Source model. [pos.pm]
psMuDeclSigma	float	mas/yr	Uncertainty of psMuDecl. [stat.error; pos.pm]
psParallax	float	mas	Stellar parallax. for the Point Source model. [pos.parallax]
psParallaxSigma	float	mas	Uncertainty of psParallax. [stat.error; pos.parallax]
uPsFlux	float	nmgy	Calibrated flux for Point Source model for u filter. [phot.count]
uPsFluxSigma	float	nmgy	Uncertainty of uPsFlux. [stat.error; phot.count]
gPsFlux	float	nmgy	Calibrated flux for Point Source model for g filter. [phot.count]
gPsFluxSigma	float	nmgy	Uncertainty of gPsFlux. [stat.error; phot.count]
rPsFlux	float	nmgy	Calibrated flux for Point Source model for r filter. [phot.count]
rPsFluxSigma	float	nmgy	Uncertainty of rPsFlux. [stat.error; phot.count]
iPsFlux	float	nmgy	Calibrated flux for Point Source model for i filter. [phot.count]
iPsFluxSigma	float	nmgy	Uncertainty of iPsFlux. [stat.error; phot.count]
zPsFlux	float	nmgy	Calibrated flux for Point Source model for z filter. [phot.count]
zPsFluxSigma	float	nmgy	Uncertainty of zPsFlux. [stat.error; phot.count]
yPsFlux	float	nmgy	Calibrated flux for Point Source model for y filter. [phot.count]
yPsFluxSigma	float	nmgy	Uncertainty of yPsFlux. [stat.error; phot.count]
psLnL	float		Natural log likelihood of the observed data given the Point Source model.
psChi2	float		Chi ² static of the model fit.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
psN	int		The number of data points (pixels) used to fit the model.
uBdRa	double	deg	RA-coordinate of the center of the object for the Bulge+Disk model at time radecTai. For u filter. [pos.eq.ra]
uBdRaSigma	float	deg	Uncertainty of uBdRa. [stat.error; pos.eq.ra]
uBdDecl	double	deg	Decl-coordinate of the center of the object for the Bulge+Disk model at time radecTai. For u filter. [pos.eq.dec]
uBdDeclSigma	float	deg	Uncertainty of uBdDecl. [stat.error; pos.eq.dec]
uBdE1	float		Ellipticity for the Bulge+Disk (e1). For u filter. [phys.size.axisRatio]
uBdE1Sigma	float		Uncertainty of uBdE1. [stat.error; phys.size.axisRatio]
uBdE2	float		Ellipticity for Bulge+Disk model (e2). For u filter. [phys.size.axisRatio]
uBdE2Sigma	float		Uncertainty of uBdE2. [stat.error; phys.size.axisRatio]
uBdFluxB	float	nmgy	Integrated flux of the de Vaucouleurs component for the Bulge+Disk model. For u filter.
uBdFluxBSigma	float	nmgy	Uncertainty of uBdFluxB.
uBdFluxD	float	nmgy	Integrated flux of the exponential component for the Bulge+Disk model. For u filter.
uBdFluxDSigma	float	nmgy	Uncertainty of uBdFluxD.
uBdReB	float	arcsec	Effective radius of the de Vaucouleurs profile component for the Bulge+Disk model. For u filter.
uBdReBSigma	float	arcsec	Uncertainty of uBdReB.
uBdReD	float	arcsec	Effective radius of the exponential profile component for the Bulge+Disk model. For u filter.
uBdReDSigma	float	arcsec	Uncertainty of uBdReD.
uBdLnL	float		Natural log likelihood of the observed data given the Bulge+Disk model. For u filter.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
uBdChi2	float		Chi ² static of the model fit. For u filter.
uBdN	int		The number of data points (pixels) used to fit the model. For u filter.
gBbdRa	double	deg	RA-coordinate of the center of the object for the Bulge+Disk model at time radecTai. For g filter. [pos.eq.ra]
gBdRaSigma	float	deg	Uncertainty of gBdRa. [stat.error; pos.eq.ra]
gBdDecl	double	deg	Decl-coordinate of the center of the object for the Bulge+Disk model at time radecTai. For g filter. [pos.eq.dec]
gBdDeclSigma	float	deg	Uncertainty of gBdDecl. [stat.error; pos.eq.dec]
gBdE1	float		Ellipticity for the Bulge+Disk (e1). For g filter. [phys.size.axisRatio]
gBdE1Sigma	float		Uncertainty of gBdE1. [stat.error; phys.size.axisRatio]
gBdE2	float		Ellipticity for Bulge+Disk model (e2). For g filter. [phys.size.axisRatio]
gBdE2Sigma	float		Uncertainty of gBdE2. [stat.error; phys.size.axisRatio]
gBdFluxB	float	nmgy	Integrated flux of the de Vaucouleurs component for the Bulge+Disk model. For g filter.
gBdFluxBSigma	float	nmgy	Uncertainty of gBdFluxB.
gBdFluxD	float	nmgy	Integrated flux of the exponential component for the Bulge+Disk model. For g filter.
gBdFluxDSigma	float	nmgy	Uncertainty of gBdFluxD.
gBdReB	float	arcsec	Effective radius of the de Vaucouleurs profile component for the Bulge+Disk model. For g filter.
gBdReBSigma	float	arcsec	Uncertainty of gBdReB.
gBdReD	float	arcsec	Effective radius of the exponential profile component for the Bulge+Disk model. For g filter.
gBdReDSigma	float	arcsec	Uncertainty of gBdReD.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
gBdLnL	float		Natural log likelihood of the observed data given the Bulge+Disk model. For g filter.
gBdChi2	float		Chi ² static of the model fit. For g filter.
gBdN	int		The number of data points (pixels) used to fit the model. For g filter.
rBdRa	double	deg	RA-coordinate of the center of the object for the Bulge+Disk model at time radecTai. For r filter. [pos.eq.ra]
rBdRaSigma	float	deg	Uncertainty of rBdRa. [stat.error; pos.eq.ra]
rBdDecl	double	deg	Decl-coordinate of the center of the object for the Bulge+Disk model at time radecTai. For r filter. [pos.eq.dec]
rBdDeclSigma	float	deg	Uncertainty of rBdDecl. [stat.error; pos.eq.dec]
rBdE1	float		Ellipticity for the Bulge+Disk (e1). For r filter. [phys.size.axisRatio]
rBdE1Sigma	float		Uncertainty of rBdE1. [stat.error; phys.size.axisRatio]
rBdE2	float		Ellipticity for Bulge+Disk model (e2). For r filter. [phys.size.axisRatio]
rBdE2Sigma	float		Uncertainty of rBdE2. [stat.error; phys.size.axisRatio]
rBdFluxB	float	nmgy	Integrated flux of the de Vaucouleurs component for the Bulge+Disk model. For r filter.
rBdFluxBSigma	float	nmgy	Uncertainty of rBdFluxB.
rBdFluxD	float	nmgy	Integrated flux of the exponential component for the Bulge+Disk model. For r filter.
rBdFluxDSigma	float	nmgy	Uncertainty of rBdFluxD.
rBdReB	float	arcsec	Effective radius of the de Vaucouleurs profile component for the Bulge+Disk model. For r filter.
rBdReBSigma	float	arcsec	Uncertainty of rBdReB.
rBdReD	float	arcsec	Effective radius of the exponential profile component for the Bulge+Disk model. For r filter.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
rBdReDSigma	float	arcsec	Uncertainty of rBdReD.
rBdLnL	float		Natural log likelihood of the observed data given the Bulge+Disk model. For r filter.
rBdChi2	float		Chi ² static of the model fit. For r filter.
rBdN	int		The number of data points (pixels) used to fit the model. For r filter.
iBdRa	double	deg	RA-coordinate of the center of the object for the Bulge+Disk model at time radecTai. For i filter. [pos.eq.ra]
iBdRaSigma	float	deg	Uncertainty of iBdRa. [stat.error; pos.eq.ra]
iBdDecl	double	deg	Decl-coordinate of the center of the object for the Bulge+Disk model at time radecTai. For i filter. [pos.eq.dec]
iBdDeclSigma	float	deg	Uncertainty of iBdDecl. [stat.error; pos.eq.dec]
iBdE1	float		Ellipticity for the Bulge+Disk (e1). For i filter. [phys.size.axisRatio]
iBdE1Sigma	float		Uncertainty of iBdE1. [stat.error; phys.size.axisRatio]
iBdE2	float		Ellipticity for Bulge+Disk model (e2). For i filter. [phys.size.axisRatio]
iBdE2Sigma	float		Uncertainty of iBdE2. [stat.error; phys.size.axisRatio]
iBdFluxB	float	nmgy	Integrated flux of the de Vaucouleurs component for the Bulge+Disk model. For i filter.
iBdFluxBSigma	float	nmgy	Uncertainty of iBdFluxB.
iBdFluxD	float	nmgy	Integrated flux of the exponential component for the Bulge+Disk model. For i filter.
iBdFluxDSigma	float	nmgy	Uncertainty of iBdFluxD.
iBdReB	float	arcsec	Effective radius of the de Vaucouleurs profile component for the Bulge+Disk model. For i filter.
iBdReBSigma	float	arcsec	Uncertainty of iBdReB.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
iBdReD	float	arcsec	Effective radius of the exponential profile component for the Bulge+Disk model. For i filter.
iBdReDSigma	float	arcsec	Uncertainty of iBdReD.
iBdLnL	float		Natural log likelihood of the observed data given the Bulge+Disk model. For i filter.
iBdChi2	float		Chi ² static of the model fit. For i filter.
iBdN	int		The number of data points (pixels) used to fit the model. For i filter.
zBbdRa	double	deg	RA-coordinate of the center of the object for the Bulge+Disk model at time radecTai. For z filter. [pos.eq.ra]
zBdRaSigma	float	deg	Uncertainty of zBdRa. [stat.error; pos.eq.ra]
zBdDecl	double	deg	Decl-coordinate of the center of the object for the Bulge+Disk model at time radecTai. For z filter. [pos.eq.dec]
zBdDeclSigma	float	deg	Uncertainty of zBdDecl. [stat.error; pos.eq.dec]
zBdE1	float		Ellipticity for the Bulge+Disk (e1). For z filter. [phys.size.axisRatio]
zBdE1Sigma	float		Uncertainty of zBdE1. [stat.error; phys.size.axisRatio]
zBdE2	float		Ellipticity for Bulge+Disk model (e2). For z filter. [phys.size.axisRatio]
zBdE2Sigma	float		Uncertainty of zBdE2. [stat.error; phys.size.axisRatio]
zBdFluxB	float	nmgy	Integrated flux of the de Vaucouleurs component for the Bulge+Disk model. For z filter.
zBdFluxBSigma	float	nmgy	Uncertainty of zBdFluxB.
zBdFluxD	float	nmgy	Integrated flux of the exponential component for the Bulge+Disk model. For z filter.
zBdFluxDSigma	float	nmgy	Uncertainty of zBdFluxD.
zBdReB	float	arcsec	Effective radius of the de Vaucouleurs profile component for the Bulge+Disk model. For z filter.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
zBdReBSigma	float	arcsec	Uncertainty of zBdReB.
zBdReD	float	arcsec	Effective radius of the exponential profile component for the Bulge+Disk model. For z filter.
zBdReDSigma	float	arcsec	Uncertainty of zBdReD.
zBdLnL	float		Natural log likelihood of the observed data given the Bulge+Disk model. For z filter.
zBdChi2	float		Chi ² static of the model fit. For z filter.
zBdN	int		The number of data points (pixels) used to fit the model. For z filter.
yBdRa	double	deg	RA-coordinate of the center of the object for the Bulge+Disk model at time radecTai. For y filter. [pos.eq.ra]
yBdRaSigma	float	deg	Uncertainty of yBdRa. [stat.error; pos.eq.ra]
yBdDecl	double	deg	Decl-coordinate of the center of the object for the Bulge+Disk model at time radecTai. For y filter. [pos.eq.dec]
yBdDeclSigma	float	deg	Uncertainty of yBdDecl. [stat.error; pos.eq.dec]
yBdE1	float		Ellipticity for the Bulge+Disk (e1). For y filter. [phys.size.axisRatio]
yBdE1Sigma	float		Uncertainty of yBdE1. [stat.error; phys.size.axisRatio]
yBdE2	float		Ellipticity for Bulge+Disk model (e2). For y filter. [phys.size.axisRatio]
yBdE2Sigma	float		Uncertainty of yBdE2. [stat.error; phys.size.axisRatio]
yBdFluxB	float	nmgy	Integrated flux of the de Vaucouleurs component for the Bulge+Disk model. For y filter.
yBdFluxBSigma	float	nmgy	Uncertainty of yBdFluxB.
yBdFluxD	float	nmgy	Integrated flux of the exponential component for the Bulge+Disk model. For y filter.
yBdFluxDSigma	float	nmgy	Uncertainty of yBdFluxD.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
yBdReB	float	arcsec	Effective radius of the de Vaucouleurs profile component for the Bulge+Disk model. For y filter.
yBdReBSigma	float	arcsec	Uncertainty of yBdReB.
yBdReD	float	arcsec	Effective radius of the exponential profile component for the Bulge+Disk model. For y filter.
yBdReDSigma	float	arcsec	Uncertainty of yBdReD.
yBdLnL	float		Natural log likelihood of the observed data given the Bulge+Disk model. For y filter.
yBdChi2	float		Chi ² static of the model fit. For y filter.
yBdN	int		The number of data points (pixels) used to fit the model. For y filter.
ugStd	float	mag	Color of the object measured in 'standard seeing'. While the exact algorithm is yet to be determined, this color is guaranteed to be seeing-independent and suitable for photo-Z determinations.
ugStdSigma	float	mag	Uncertainty of ugStd.
grStd	float	mag	Color of the object measured in 'standard seeing'. While the exact algorithm is yet to be determined, this color is guaranteed to be seeing-independent and suitable for photo-Z determinations.
grStdSigma	float	mag	Uncertainty of grStd.
riStd	float	mag	Color of the object measured in 'standard seeing'. While the exact algorithm is yet to be determined, this color is is guaranteed to be seeing-independent and suitable for photo-Z determinations.
riStdSigma	float	mag	Uncertainty of riStd.
izStd	float	mag	Color of the object measured in 'standard seeing'. While the exact algorithm is yet to be determined, this color is guaranteed to be seeing-independent and suitable for photo-Z determinations.
izStdSigma	float	mag	Uncertainty of izStd.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
zyStd	float	mag	Color of the object measured in 'standard seeing'. While the exact algorithm is yet to be determined, this color is guaranteed to be seeing-independent and suitable for photo-Z determinations.
zyStdSigma	float	mag	Uncertainty of zyStd.
uRa	double	arcsec	RA-coordinate coordinate of the centroid computed for u filter. [pos.eq.ra]
uRaSigma	double	arcsec	Uncertainty of uRa. [stat.error; pos.eq.ra]
uDecl	double	arcsec	Decl-coordinate coordinate of the centroid computed for u filter. [pos.eq.dec]
uDeclSigma	double	arcsec	Uncertainty of uDecl. [stat.error; pos.eq.dec]
gRa	double	arcsec	RA-coordinate coordinate of the centroid computed for g filter. [pos.eq.ra]
gRaSigma	double	arcsec	Uncertainty of gRa. [stat.error; pos.eq.ra]
gDecl	double	arcsec	Decl-coordinate coordinate of the centroid computed for g filter. [pos.eq.dec]
gDeclSigma	double	arcsec	Uncertainty of gDecl. [stat.error; pos.eq.dec]
rRa	double	arcsec	RA-coordinate coordinate of the centroid computed for r filter. [pos.eq.ra]
rRaSigma	double	arcsec	Uncertainty of rRa. [stat.error; pos.eq.ra]
rDecl	double	arcsec	Decl-coordinate coordinate of the centroid computed for r filter. [pos.eq.dec]
rDeclSigma	double	arcsec	Uncertainty of rDecl. [stat.error; pos.eq.dec]
iRa	double	arcsec	RA-coordinate coordinate of the centroid computed for i filter. [pos.eq.ra]
iRaSigma	double	arcsec	Uncertainty of iRa. [stat.error; pos.eq.ra]
iDecl	double	arcsec	Decl-coordinate coordinate of the centroid computed for i filter. [pos.eq.dec]
iDeclSigma	double	arcsec	Uncertainty of iDecl. [stat.error; pos.eq.dec]
zRa	double	arcsec	RA-coordinate coordinate of the centroid computed for z filter. [pos.eq.ra]
zRaSigma	double	arcsec	Uncertainty of zRa. [stat.error; pos.eq.ra]

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
zDecl	double	arcsec	Decl-coordinate coordinate of the centroid computed for z filter. [pos.eq.dec]
zDeclSigma	double	arcsec	Uncertainty of zDecl. [stat.error; pos.eq.dec]
yRa	double	arcsec	RA-coordinate coordinate of the centroid computed for y filter. [pos.eq.ra]
yRaSigma	double	arcsec	Uncertainty of yRa. [stat.error; pos.eq.ra]
yDecl	double	arcsec	Decl-coordinate coordinate of the centroid computed for y filter. [pos.eq.dec]
yDeclSigma	double	arcsec	Uncertainty of yDecl. [stat.error; pos.eq.dec]
uE1	float		Adaptive e1 shape measure for u filter. See Bernstein and Jarvis (2002) for detailed discussion of all adaptive-moment related quantities. [phys.size.axisRatio]
uE1Sigma	float		Uncertainty of uE1 [stat.error; phys.size.axisRatio]
uE2	float		Adaptive e2 shape measure for u filter. See Bernstein and Jarvis (2002) for detailed discussion of all adaptive-moment related quantities. [phys.size.axisRatio]
uE2Sigma	float		Uncertainty of uE2 [stat.error; phys.size.axisRatio]
uE1_E2_Cov	float		Covariance of uE1 and uE2.
gE1	float		Adaptive e1 shape measure for g filter. See Bernstein and Jarvis (2002) for detailed discussion of all adaptive-moment related quantities. [phys.size.axisRatio]
gE1Sigma	float		Uncertainty of gE1 [stat.error; phys.size.axisRatio]
gE2	float		Adaptive e2 shape measure for g filter. See Bernstein and Jarvis (2002) for detailed discussion of all adaptive-moment related quantities. [phys.size.axisRatio]
gE2Sigma	float		Uncertainty of gE2 [stat.error; phys.size.axisRatio]
gE1_E2_Cov	float		Covariance of gE1 and gE2.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
rE1	float		Adaptive e1 shape measure for r filter. See Bernstein and Jarvis (2002) for detailed discussion of all adaptive-moment related quantities. [phys.size.axisRatio]
rE1Sigma	float		Uncertainty of rE1 [stat.error; phys.size.axisRatio]
rE2	float		Adaptive e2 shape measure for r filter. See Bernstein and Jarvis (2002) for detailed discussion of all adaptive-moment related quantities. [phys.size.axisRatio]
rE2Sigma	float		Uncertainty of rE2 [stat.error; phys.size.axisRatio]
rE1_E2_Cov	float		Covariance of rE1 and rE2.
iE1	float		Adaptive e1 shape measure for i filter. See Bernstein and Jarvis (2002) for detailed discussion of all adaptive-moment related quantities. [phys.size.axisRatio]
iE1Sigma	float		Uncertainty of iE1 [stat.error; phys.size.axisRatio]
iE2	float		Adaptive e2 shape measure for i filter. See Bernstein and Jarvis (2002) for detailed discussion of all adaptive-moment related quantities. [phys.size.axisRatio]
iE2Sigma	float		Uncertainty of iE2 [stat.error; phys.size.axisRatio]
iE1_E2_Cov	float		Covariance of iE1 and iE2.
zE1	float		Adaptive e1 shape measure for z filter. See Bernstein and Jarvis (2002) for detailed discussion of all adaptive-moment related quantities. [phys.size.axisRatio]
zE1Sigma	float		Uncertainty of zE1 [stat.error; phys.size.axisRatio]
zE2	float		Adaptive e2 shape measure for z filter. See Bernstein and Jarvis (2002) for detailed discussion of all adaptive-moment related quantities. [phys.size.axisRatio]
zE2Sigma	float		Uncertainty of zE2 [stat.error; phys.size.axisRatio]

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
zE1_E2_Cov	float		Covariance of zE1 and zE2.
yE1	float		Adaptive e1 shape measure for y filter. See Bernstein and Jarvis (2002) for detailed discussion of all adaptive-moment related quantities. [phys.size.axisRatio]
yE1Sigma	float		Uncertainty of yE1 [stat.error; phys.size.axisRatio]
yE2	float		Adaptive e2 shape measure for y filter. See Bernstein and Jarvis (2002) for detailed discussion of all adaptive-moment related quantities. [phys.size.axisRatio]
yE2Sigma	float		Uncertainty of yE2 [stat.error; phys.size.axisRatio]
yE1_E2_Cov	float		Covariance of yE1 and yE2.
uMSum	float		Sum of second adaptive moments for u filter.
uMSumSigma	float		Uncertainty of uMSum
gMSum	float		Sum of second adaptive moments for g filter.
gMSumSigma	float		Uncertainty of gMSum
rMSum	float		Sum of second adaptive moments for r filter.
rMSumSigma	float		Uncertainty of rMSum
iMSum	float		Sum of second adaptive moments for i filter.
iMSumSigma	float		Uncertainty of iMSum
zMSum	float		Sum of second adaptive moments for z filter.
zMSumSigma	float		Uncertainty of zMSum
yMSum	float		Sum of second adaptive moments for y filter.
yMSumSigma	float		Uncertainty of yMSum
uM4	float		Fourth order adaptive moment for u filter.
gM4	float		Fourth order adaptive moment for g filter.
rM4	float		Fourth order adaptive moment for r filter.
iM4	float		Fourth order adaptive moment for i filter.
zM4	float		Fourth order adaptive moment for z filter.
yM4	float		Fourth order adaptive moment for y filter.
uPetroRad	float	arcsec	Petrosian radius, computed using elliptical apertures defined by the adaptive moments for u filter.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
uPetroRadSigma	float	arcsec	Uncertainty of uPetroRad
gPetroRad	float	arcsec	Petrosian radius, computed using elliptical apertures defined by the adaptive moments for g filter.
gPetroRadSigma	float	arcsec	Uncertainty of gPetroRad
rPetroRad	float	arcsec	Petrosian radius, computed using elliptical apertures defined by the adaptive moments for r filter.
rPetroRadSigma	float	arcsec	Uncertainty of rPetroRad
iPetroRad	float	arcsec	Petrosian radius, computed using elliptical apertures defined by the adaptive moments for i filter.
iPetroRadSigma	float	arcsec	Uncertainty of iPetroRad
zPetroRad	float	arcsec	Petrosian radius, computed using elliptical apertures defined by the adaptive moments for z filter.
zPetroRadSigma	float	arcsec	Uncertainty of zPetroRad
yPetroRad	float	arcsec	Petrosian radius, computed using elliptical apertures defined by the adaptive moments for y filter.
yPetroRadSigma	float	arcsec	Uncertainty of yPetroRad
petroFilter	char		Name of the filter of the canonical petroRad.
uPetroFlux	float	nmgy	Petrosian flux within a defined multiple of the canonical petroRad for u filter.
uPetroFluxSigma	float	nmgy	Uncertainty of uPetroFlux.
gPetroFlux	float	nmgy	Petrosian flux within a defined multiple of the canonical petroRad for g filter.
gPetroFluxSigma	float	nmgy	Uncertainty of gPetroFlux.
rPetroFlux	float	nmgy	Petrosian flux within a defined multiple of the canonical petroRad for r filter.
rPetroFluxSigma	float	nmgy	Uncertainty of rPetroFlux.
iPetroFlux	float	nmgy	Petrosian flux within a defined multiple of the canonical petroRad for i filter.
iPetroFluxSigma	float	nmgy	Uncertainty of iPetroFlux.
zPetroFlux	float	nmgy	Petrosian flux within a defined multiple of the canonical petroRad for z filter.
zPetroFluxSigma	float	nmgy	Uncertainty of zPetroFlux.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
yPetroFlux	float	nmgy	Petrosian flux within a defined multiple of the canonical petroRad for y filter.
yPetroFluxSigma	float	nmgy	Uncertainty of yPetroFlux.
uPetroRad50	float	arcsec	Radius containing 50% of Petrosian flux for u filter.
uPetroRad50Sigma	float	arcsec	Uncertainty of uPetroRad50.
gPetroRad50	float	arcsec	Radius containing 50% of Petrosian flux for g filter.
gPetroRad50Sigma	float	arcsec	Uncertainty of gPetroRad50.
rPetroRad50	float	arcsec	Radius containing 50% of Petrosian flux for r filter.
rPetroRad50Sigma	float	arcsec	Uncertainty of rPetroRad50.
iPetroRad50	float	arcsec	Radius containing 50% of Petrosian flux for i filter.
iPetroRad50Sigma	float	arcsec	Uncertainty of iPetroRad50.
zPetroRad50	float	arcsec	Radius containing 50% of Petrosian flux for z filter.
zPetroRad50Sigma	float	arcsec	Uncertainty of zPetroRad50.
yPetroRad50	float	arcsec	Radius containing 50% of Petrosian flux for y filter.
yPetroRad50Sigma	float	arcsec	Uncertainty of yPetroRad50.
uPetroRad90	float	arcsec	Radius containing 90% of Petrosian flux for u filter.
uPetroRad90Sigma	float	arcsec	Uncertainty of uPetroRad90.
gPetroRad90	float	arcsec	Radius containing 90% of Petrosian flux for g filter.
gPetroRad90Sigma	float	arcsec	Uncertainty of gPetroRad90.
rPetroRad90	float	arcsec	Radius containing 90% of Petrosian flux for r filter.
rPetroRad90Sigma	float	arcsec	Uncertainty of rPetroRad90.
iPetroRad90	float	arcsec	Radius containing 90% of Petrosian flux for i filter.
iPetroRad90Sigma	float	arcsec	Uncertainty of iPetroRad90.
zPetroRad90	float	arcsec	Radius containing 90% of Petrosian flux for z filter.
zPetroRad90Sigma	float	arcsec	Uncertainty of zPetroRad90.
yPetroRad90	float	arcsec	Radius containing 90% of Petrosian flux for y filter.
yPetroRad90Sigma	float	arcsec	Uncertainty of yPetroRad90.
uKronRad	float	arcsec	Kron radius (computed using elliptical apertures defined by the adaptive moments) for u filter.
uKronRadSigma	float	arcsec	Uncertainty of uKronRad.
gKronRad	float	arcsec	Kron radius (computed using elliptical apertures defined by the adaptive moments) for g filter.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
gKronRadSigma	float	arcsec	Uncertainty of gKronRad.
rKronRad	float	arcsec	Kron radius (computed using elliptical apertures defined by the adaptive moments) for r filter.
rKronRadSigma	float	arcsec	Uncertainty of rKronRad.
iKronRad	float	arcsec	Kron radius (computed using elliptical apertures defined by the adaptive moments) for i filter.
iKronRadSigma	float	arcsec	Uncertainty of iKronRad.
zKronRad	float	arcsec	Kron radius (computed using elliptical apertures defined by the adaptive moments) for z filter.
zKronRadSigma	float	arcsec	Uncertainty of zKronRad.
yKronRad	float	arcsec	Kron radius (computed using elliptical apertures defined by the adaptive moments) for y filter.
yKronRadSigma	float	arcsec	Uncertainty of yKronRad.
kronFilter	char		The filter of the canonical kronRad.
uKronFlux	float	nmgy	Kron flux within a defined multiple of the canonical kronRad for u filter.
uKronFluxSigma	float	nmgy	Uncertainty of uKronFlux.
gKronFlux	float	nmgy	Kron flux within a defined multiple of the canonical kronRad for g filter.
gKronFluxSigma	float	nmgy	Uncertainty of gKronFlux.
rKronFlux	float	nmgy	Kron flux within a defined multiple of the canonical kronRad for r filter.
rKronFluxSigma	float	nmgy	Uncertainty of rKronFlux.
iKronFlux	float	nmgy	Kron flux within a defined multiple of the canonical kronRad for i filter.
iKronFluxSigma	float	nmgy	Uncertainty of iKronFlux.
zKronFlux	float	nmgy	Kron flux within a defined multiple of the canonical kronRad for z filter.
zKronFluxSigma	float	nmgy	Uncertainty of zKronFlux.
yKronFlux	float	nmgy	Kron flux within a defined multiple of the canonical kronRad for y filter.
yKronFluxSigma	float	nmgy	Uncertainty of yKronFlux.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
uKronRad50	float	arcsec	Radius containing 50% of Kron flux for u filter.
uKronRad50Sigma	float	arcsec	Uncertainty of uKronRad50.
gKronRad50	float	arcsec	Radius containing 50% of Kron flux for g filter.
gKronRad50Sigma	float	arcsec	Uncertainty of gKronRad50.
rKronRad50	float	arcsec	Radius containing 50% of Kron flux for r filter.
rKronRad50Sigma	float	arcsec	Uncertainty of rKronRad50.
iKronRad50	float	arcsec	Radius containing 50% of Kron flux for i filter.
iKronRad50Sigma	float	arcsec	Uncertainty of iKronRad50.
zKronRad50	float	arcsec	Radius containing 50% of Kron flux for z filter.
zKronRad50Sigma	float	arcsec	Uncertainty of zKronRad50.
yKronRad50	float	arcsec	Radius containing 50% of Kron flux for y filter.
yKronRad50Sigma	float	arcsec	Uncertainty of yKronRad50.
uKronRad90	float	arcsec	Radius containing 90% of Kron flux for u filter.
uKronRad90Sigma	float	arcsec	Uncertainty of uKronRad90.
gKronRad90	float	arcsec	Radius containing 90% of Kron flux for g filter.
gKronRad90Sigma	float	arcsec	Uncertainty of gKronRad90.
rKronRad90	float	arcsec	Radius containing 90% of Kron flux for r filter.
rKronRad90Sigma	float	arcsec	Uncertainty of rKronRad90.
iKronRad90	float	arcsec	Radius containing 90% of Kron flux for i filter.
iKronRad90Sigma	float	arcsec	Uncertainty of iKronRad90.
zKronRad90	float	arcsec	Radius containing 90% of Kron flux for z filter.
zKronRad90Sigma	float	arcsec	Uncertainty of zKronRad90.
yKronRad90	float	arcsec	Radius containing 90% of Kron flux for y filter.
yKronRad90Sigma	float	arcsec	Uncertainty of yKronRad90.
uApN	byte		Number of elliptical annuli (see below) for u filter.
gApN	byte		Number of elliptical annuli (see below) for g filter.
rApN	byte		Number of elliptical annuli (see below) for r filter.
iApN	byte		Number of elliptical annuli (see below) for i filter.
zApN	byte		Number of elliptical annuli (see below) for z filter.
yApN	byte		Number of elliptical annuli (see below) for y filter.

Continued on next page

Table 7: Object Table

Column Name	Data type	Unit	Description [UCD]
extendedness	float		A measure of extendedness, computed using a combination of available moments and model fluxes or from a likelihood ratio of point/trailed source models (exact algorithm TBD). extendedness = 1 implies a high degree of confidence that the source is extended. extendedness = 0 implies a high degree of confidence that the source is point-like.
FLAGS1	long		Flags, tbd.
FLAGS2	long		Flags, tbd.

7.8 Object_Extra

Less frequently used information from The Object table.

Table 8: Object_Extra Table

Column Name	Data type	Unit	Description [UCD]
objectId	long		Unique id. [meta.id; src]
psCov	binary		Various covariances for Point Source model. 66 TINYINTs.
bdCov	binary		Covariance matrix for the Bulge+Disk model. 168 TINYINTs. $(((8 \times 9/2) - 8) * 6)$.
bdSamples	binary		Independent samples of Bulge+Disk likelihood surface. All sampled quantities will be stored with at least 3 significant digits of precision. The number of samples will vary from object to object, depending on how well the object's likelihood function is approximated by a Gaussian. We are assuming on average $[9 \times 200 \times 4 \text{ FLOAT16}]$.

Continued on next page

Table 8: Object_Extra Table

Column Name	Data type	Unit	Description [UCD]
photoZ	binary		Photometric redshift likelihood samples (pairs of z , $\log L$) computed using a to-be-determined published and widely accepted algorithm at the time of LSST Commissioning. [2x100 FLOAT].

7.9 Object_APMean

Aperture mean (per bin) for the Object table. We expect 8 bins on average per object.

Table 9: Object_APMean Table

Column Name	Data type	Unit	Description [UCD]
objectId	long		Unique id. [meta.id; src]
filterName	char		Name of the filter.
binN	byte		A bin in radius at which the aperture measurement is being performed.
sbMean	float	nmgy/arcsec	Mean surface brightness at which the aperture measurement is being performed.
sbSigma	float		Standard deviation of pixel surface brightness in annulus.

7.10 Object_Periodic

Definition of periodic features for Object table. We expect about 32 per object.

Table 10: Object_Periodic Table

Column Name	Data type	Unit	Description [UCD]
objectId	long		Unique id. [meta.id; src]
filterName	char		Name of the filter.
theFeature	byte		Feature/metric identifier.

Continued on next page

Table 10: Object_Periodic Table

Column Name	Data type	Unit	Description [UCD]
theValue	float		Feature/metric value.

7.11 Object_NonPeriodic

Definition of non-periodic features for Object table. We expect about 20 per object.

Table 11: Object_NonPeriodic Table

Column Name	Data type	Unit	Description [UCD]
objectId	long		Unique id. [meta.id; src]
filterName	char		Name of the filter.
theFeature	byte		Feature/metric identifier.
theValue	float		Feature/metric value.

7.12 Source

Table to store high signal-to-noise “sources”. A source is a measurement of Object’s properties from a single image that contains its footprint on the sky.

Table 12: Source Table

Column Name	Data type	Unit	Description [UCD]
sourceId	long		Unique id. [meta.id; src]
ccdVisitId	long		Id of the ccdVisit where this source was measured. Note that we are allowing a source to belong to multiple amplifiers, but it may not span multiple ccds. [meta.id; obs.image]
filterName	char		Name of the filter used to take the two exposures where this source was measured. [meta.id; instr.filter]

Continued on next page

Table 12: Source Table

Column Name	Data type	Unit	Description [UCD]
objectId	long		Id of the corresponding object. Note that this might be NULL (each source will point to either object or ssObject). [meta.id; src]
ssObjectId	long		Id of the corresponding ssObject. Note that this might be NULL (each source will point to either object or ssObject). [meta.id; src]
parentSourceId	long		Id of the parent source this source has been deblended from, if any.
htmlId20	long		HTM index.
sky	float	nmgy/asec ²	Estimated sky background at the position (centroid) of the source.
skySigma	float	nmgy/asec ²	Uncertainty of sky.
psFlux	float	nmgy	Calibrated flux for Point Source model. [phot.count]
psFluxSigma	float	nmgy	Uncertainty of psFlux.
psX	float	pixels	Point source model (x) position of the object on the CCD.
psXSigma	float	pixels	Uncertainty of psX.
psY	float		Point source model (y) position of the object on the CCD.
psYSigma	float	pixels	Uncertainty of psY.
psFlux_psX_Cov	float		Covariance of psFlux and psX.
psFlux_psY_Cov	float		Covariance of psFlux and psY.
psX_Y_Cov	float		Covariance of psX and psY.
psLnL	float		Natural log likelihood of the observed data given the Point Source model.
psChi2	float		Chi ² static of the model fit.
psN	int		The number of data points (pixels) used to fit the model.
psRa	double	deg	RA-coordinate of the center of the object for the Point Source model at time radecTai. [pos.eq.ra]
psRaSigma	float	deg	Uncertainty of psRa. [stat.error; pos.eq.ra]

Continued on next page

Table 12: Source Table

Column Name	Data type	Unit	Description [UCD]
psDecl	double	deg	Decl-coordinate of the center of the object for the Point Source model at time radecTai. [pos.eq.dec]
psDeclSigma	float	deg	Uncertainty of psDecl. [stat.error; pos.eq.dec]
psFlux_psRa_Cov	float		Covariance of psFlux and psRa.
psFlux_psDecl_Cov	float		Covariance of psFlux and psRa.
x	float	pixel	x position computed using an algorithm similar to that used by SDSS. [pos.cartesian.x]
xSigma	float	pixel	Uncertainty of x. [stat.error; pos.cartesian.x]
y	float	pixel	y position computed using an algorithm similar to that used by SDSS. [pos.cartesian.y]
ySigma	float	pixel	Uncertainty of y. [stat.error; pos.cartesian.y]
x_y_Cov	float	pixel ²	Covariance between x and y.
ra	double	arcsec	Calibrated RA-coordinate of the center of the source transformed from xy. [pos.eq.ra]
raSigma	float	arcsec	Uncertainty of ra. [stat.error; pos.eq.ra]
decl	double	arcsec	Calibrated Decl-coordinate of the center of the source transformed from xy. [pos.eq.dec]
declSigma	float	arcsec	Uncertainty of decl. [stat.error; pos.eq.dec]
ra_decl_Cov	float	arcsec ²	Covariance of ra and decl.
E1	float		Adaptive e1 shape measure of the source as measured on the difference image. [phys.size.axisRatio]
E1Sigma	float		Uncertainty of E1. [stat.error; phys.size.axisRatio]
E2	float		Adaptive e2 shape measure of the source as measured on the difference image. [phys.size.axisRatio]
E2Sigma	float		Uncertainty of E2. [stat.error; phys.size.axisRatio]
E1_E2_Cov	float		Covariance of E1 and E2
mSum	float		Sum of second adaptive moments.
mSumSigma	float		Uncertainty of mSum.
m4	float		Fourth order adaptive moment.
apN	byte		Number of elliptical annuli.

Continued on next page

Table 12: Source Table

Column Name	Data type	Unit	Description [UCD]
flags	long		Flags. Tbd.

7.13 Source_APMean

Aperture mean (per bin) for the Source table. We expect 8 bins on average per source.

Table 13: Source_APMean Table

Column Name	Data type	Unit	Description [UCD]
sourceId	long		Unique id. [meta.id; src]
binN	byte		A bin in radius at which the aperture measurement is being performed.
sbMean	float	nmgy/arcsec	Mean surface brightness at which the aperture measurement is being performed.
sbSigma	float		Standard deviation of pixel surface brightness in annulus.

7.14 ForcedSource

Forced-photometry source measurement on an individual Exposure based on a Multifit shape model derived from a deep detection.

Table 14: ForcedSource Table

Column Name	Data type	Unit	Description [UCD]
objectId	long		[meta.id; src]
ccdVisitId	long		Id of the ccd visit where this forcedSource was measured. Note that we are allowing a forced-Source to belong to multiple amplifiers, but it may not span multiple ccDs. [meta.id; obs.image]
psFlux	float	nmgy	Point Source model flux. [phot.count]

Continued on next page

Table 14: ForcedSource Table

Column Name	Data type	Unit	Description [UCD]
psFlux_Sigma	float	nmgy	Uncertainty of psFlux. [stat.error; phot.count]
flags	byte		Flags, bitwise OR tbd [meta.code]

7.15 RawAmpExposure

Exposure for one amplifier (raw image).

Table 15: RawAmpExposure Table

Column Name	Data type	Unit	Description [UCD]
rawAmpExposureId	long		Primary key (unique identifier). [meta.id; obs.image]
rawCcdExposureId	long		Pointer to RawCcdExposure containing this amp exposure. [meta.id; obs.image]
ampName	char		Amplifier name. [meta.id; instr.det]
ccdName	char		Ccd name. [meta.id; instr.det]
raftName	char		Raft name. [meta.id; instr.det]
filterName	char		Filter name. [meta.id; instr.filter]
ra	double	deg	Ra of amp center. [pos.eq.ra]
decl	double	deg	Decl of amp center. [pos.eq.dec]
skyRotation	double	deg	Sky rotation angle.
bias	float	DN	Bias as measured from overscan columns.
biasNom	float	DN	Preset bias.
gain	float	electrons/DN	Measured gain value.
gainNom	float	DN	Preset gain.
rdNoise	float	DN	Read noise for detector/amplifier.
saturation	int	DN	Maximum data value for A/D converter.
llcx	int	pixels	FITS pixel coordinates (0.5, 0.5).
llcy	int	pixels	FITS pixel coordinates (0.5, 0.5).
ulcx	int	pixels	FITS pixel coordinates (0.5, 0.5+NAXIS2).
ulcy	int	pixels	FITS pixel coordinates (0.5, 0.5+NAXIS2).
urcx	int	pixels	FITS pixel coordinates (0.5+NAXIS1, 0.5+NAXIS2).

Continued on next page

Table 15: RawAmpExposure Table

Column Name	Data type	Unit	Description [UCD]
urcy	int	pixels	FITS pixel coordinates (0.5+NAXIS1, 0.5+NAXIS2).
lrcx	int	pixels	FITS pixel coordinates (0.5+NAXIS1, 0.5).
lrcy	int	pixels	FITS pixel coordinates (0.5+NAXIS1, 0.5).
xSize	int		Number of columns in the image.
ySize	int		Number of rows in the image.
obsStart	timestamp		Start of the exposure, TAI, accurate to 10ms. [time.start]
expMidpt	double		Midpoint for exposure. TAI, accurate to 10ms. [time.epoch]
expTime	double	s	Duration of exposure, accurate to 10ms. [time.duration]
darkTime	double	s	Dark current accumulation time, accurate to 10ms. [time.duration]
flags	int		[meta.code]

7.16 RawAmpExposureMetadata

Generic key-value pair metadata for RawAmpExposure.

Table 16: RawAmpExposureMetadata Table

Column Name	Data type	Unit	Description [UCD]
rawAmpExposureId	long		Id of the corresponding RawAmpExposure. [meta.id; obs.image]
metadataKey	string		
intValue	int		
doubleValue	double		
stringValue	string		

7.17 RawCcdExposure

Table 17: RawCcdExposure Table

Column Name	Data type	Unit	Description [UCD]
rawCcdExposureId	long		Primary key (unique identifier). [meta.id; obs.image]
rawExposureId	long		Point to the RawExposure containing this ccd exposure. [meta.id; obs.image]
ccdName	char		Ccd name. [meta.id; instr.det]
raftName	char		Raft name. [meta.id; instr.det]
filterName	char		Filter name. [meta.id; instr.filter]
ra	double	deg	RA of field of view center. [pos.eq.ra]
decl	double	deg	Declination of field of view center. [pos.eq.dec]
skyRotation	double	deg	Sky rotation angle.
zenithDistance	float	deg	Zenith distance at observation mid-point. [pos.az.zd]
airmass	float		Airmass at the observed line of sight.
llcx	int	pixels	FITS pixel coordinates (0.5, 0.5).
llcy	int	pixels	FITS pixel coordinates (0.5, 0.5).
ulcx	int	pixels	FITS pixel coordinates (0.5, 0.5+ySize).
ulcy	int	pixels	FITS pixel coordinates (0.5, 0.5+ySize).
urcx	int	pixels	FITS pixel coordinates (0.5+xSize, 0.5+ySize).
urcy	int	pixels	FITS pixel coordinates (0.5+xSize, 0.5+ySize).
lrcx	int	pixels	FITS pixel coordinates (0.5+xSize, 0.5).
lrcy	int	pixels	FITS pixel coordinates (0.5+xSize, 0.5).
xSize	int		Number of rows in the image.
ySize	int		Number of columns in the image.
obsStart	timestamp		Start of the exposure, TAI, accurate to 10ms. [time.start]
expMidpt	double		Midpoint for exposure. TAI, accurate to 10ms. [time.epoch]
expTime	double	s	Duration of exposure, accurate to 10ms. [time.duration]
darkTime	double	s	Dark current accumulation time, accurate to 10ms. [time.duration]
ccdTemp	float		Temperature measured on the Ccd.

Continued on next page

Table 17: RawCcdExposure Table

Column Name	Data type	Unit	Description [UCD]
binX	int	pixel	Binning of the ccd in x (row) direction. [meta.number]
binY	int	pixel	Binning of the ccd in y (column) direction. [meta.number]
WCS	binary		A nominal WCS derives from telescope pointing information (not fitted). [10x8 BYTES].
flags	int		Flags, bitwise OR tbd [meta.code]

7.18 RawCcdExposureMetadata

Generic key-value pair metadata for RawCcdExposure.

Table 18: RawCcdExposureMetadata Table

Column Name	Data type	Unit	Description [UCD]
rawCcdExposureId	long		Id of the corresponding RawCcdExposure. [meta.id; obs.image]
metadataKey	string		
metadataValue	string		

7.19 RawExposure

Raw exposure (entire exposure, all ccds).

Table 19: RawExposure Table

Column Name	Data type	Unit	Description [UCD]
rawExposureId	long		Primary key (unique identifier.) [meta.id; obs.image]
filterName	char		Filter name. [meta.id; instr.filter]
ra	double	deg	Right Ascension of focal plane center. [pos.eq.ra]

Continued on next page

Table 19: RawExposure Table

Column Name	Data type	Unit	Description [UCD]
decl	double	deg	Declination of focal plane center. [pos.eq.dec]
skyRotation	double	deg	Sky rotation angle.
altitude	double	deg	Altitude of focal plane center.
azimuth	double	deg	Azimuth of focal plane center.
programId	int		Observing program id (e.g., universal cadence, or one of the deep drilling programs, etc.).
exposureType	byte		Type of exposure (science exposure, dark, flat, etc.).
zenithDistance	float	deg	Zenith distance at observation mid-point. [pos.az.zd]
airmass	float		Airmass of the observed line of sight.
obsStart	timestamp		Start of the exposure, TAI, accurate to 10ms. [time.start]
expMidpt	double		Midpoint for exposure. TAI, accurate to 10ms. [time.epoch]
expTime	double	s	Duration of exposure, accurate to 10ms. [time.duration]
cameraTemp	float	K	Temperature measured of the camera.
mirror1Temp	float	K	Primary mirror temperature.
mirror2Temp	float	K	Secondary mirror temperature.
mirror3Temp	float	K	Tertiary mirror temperature.
domeTemp	float	K	Dome temperature.
externalTemp	float	K	Temperature outside the dome.
dimSeeing	float	arcsec	Seeing measured by the differential image motion monitor.
pwwGPS	float	mm	GPS-based measurement of precipitable water vapor (PVW).
pwwMW	float	mm	Microwave radiometer measurement of PVW.
flags	int		Flags, bitwise OR tbd [meta.code]

7.20 CcdVisit

Table 20: CcdVisit Table

Column Name	Data type	Unit	Description [UCD]
ccdVisitId	long		Primary key (unique identifier). [meta.id; obs.image]
visitId	int		Reference to the corresponding entry in the Visit table. [meta.id; obs.exposure]
ccdName	char		Ccd name. [meta.id; instr.det]
raftName	char		Raft name. [meta.id; instr.det]
filterName	char		Filter name used for this exposure. [meta.id; instr.filter]
nExposures	int		Number of exposures combined to produce this visit.
ra	double	deg	RA of Ccd center. [pos.eq.ra]
decl	double	deg	Decl of Ccd center. [pos.eq.dec]
skyRotation	double	deg	Sky rotation angle.
zenithDistance	float	deg	Zenith distance at observation mid-point. [pos.az.zd]
llcx	int	pixels	FITS pixel coordinates (0.5, 0.5).
llcy	int	pixels	FITS pixel coordinates (0.5, 0.5).
ulcx	int	pixels	FITS pixel coordinates (0.5, 0.5+ySize).
ulcy	int	pixels	FITS pixel coordinates (0.5, 0.5+ySize).
urcx	int	pixels	FITS pixel coordinates (0.5+xSize, 0.5+ySize).
urcy	int	pixels	FITS pixel coordinates (0.5+xSize, 0.5+ySize).
lrcx	int	pixels	FITS pixel coordinates (0.5+xSize, 0.5).
lrcy	int	pixels	FITS pixel coordinates (0.5+xSize, 0.5).
xSize	int		Number of columns in the image.
ySize	int		Number of rows in the image.
obsStart	timestamp		Start of the exposure, TAI, accurate to 10ms. [time.start]
expMidpt	double		Midpoint for exposure. TAI, accurate to 10ms. [time.epoch]
expTime	double	s	Average duration of exposure, accurate to 10ms. [time.duration]

Continued on next page

Table 20: CcdVisit Table

Column Name	Data type	Unit	Description [UCD]
darkTime	double	s	Average dark current accumulation time, accurate to 10ms. [time.duration]
ccdTemp	float		Temperature measured on the Ccd.
binX	int	pixel	Binning of the ccd in x (row) direction. [meta.number]
binY	int	pixel	Binning of the ccd in y (column) direction. [meta.number]
WCS	binary		Precise WCS solution for the Ccd. [100x4 BYTES].
zeroPoint	float		Zero-point for the Ccd, estimated at Ccd center.
seeing	float	arcsec	Mean measured FWHM of the PSF.
skyBg	float	DN	Average sky background.
skyNoise	float	DN	RMS noise of the sky background.
flags	int		Flags, bitwise OR tbd [meta.code]

7.21 CcdVisitMetadata

Generic key-value pair metadata for CcdVisit.

Table 21: CcdVisitMetadata Table

Column Name	Data type	Unit	Description [UCD]
ccdVisitId	long		[meta.id; obs.image]
metadataKey	string		
intValue	int		
doubleValue	double		
stringValue	string		

7.22 Visit

Defines a single Visit.

Table 22: Visit Table

Column Name	Data type	Unit	Description [UCD]
visitId	int		Unique identifier. [meta.id; obs.image]
filterName	char		Filter name. [meta.id; instr.filter]
nExposures	int		Number of exposures combined to produce this visit.
ra	double	deg	RA of focal plane center. [pos.eq.ra]
decl	double	deg	Decl of focal plane center. [pos.eq.dec]
skyRotation	double	deg	Sky rotation angle.
altitude	double	deg	Altitude of focal plane center.
azimuth	double	deg	Azimuth of focal plane center.
programId	int		Observing program id (e.g., universal cadence, or one of the deep drilling programs, etc.).
exposureType	byte		Type of exposure (science exposure, dark, flat, etc.).
zenithDistance	float	deg	Zenith distance at observation mid-point. [pos.az.zd]
airmass	float		Airmass of the observed line of sight.
obsStart	timestamp		Start of the exposure at the fiducial center of the focal plane array, TAI, accurate to 10ms. [time.start]
expMidpt	double		Midpoint for exposure at the fiducial center of the focal plane array. TAI, accurate to 10ms. [time.epoch]
expTime	double	s	Average duration of exposure, accurate to 10ms. [time.duration]
cameraTemp	float	K	Temperature measured of the camera.
mirror1Temp	float	K	Primary mirror temperature.
mirror2Temp	float	K	Secondary mirror temperature.
mirror3Temp	float	K	Tertiary mirror temperature.
domeTemp	float	K	Dome temperature.
externalTemp	float	K	Temperature outside the dome.
dimSeeing	float	arcsec	Seeing measured by the differential image motion monitor.

Continued on next page

Table 22: Visit Table

Column Name	Data type	Unit	Description [UCD]
pwwGPS	float	mm	GPS-based measurement of precipitable water vapor (PVW).
pwwMW	float	mm	Microwave radiometer measurement of PVW.
flags	int		Flags, bitwise OR tbd [meta.code]

7.23 VisitMetadata

Visit-related generic key-value pair metadata.

Table 23: VisitMetadata Table

Column Name	Data type	Unit	Description [UCD]
visitId	int		Id of the corresponding Visit. [meta.id; obs.image]
metadataKey	string		
metadataValue	string		

7.24 Visit_To_RawExposure

Mapping table: Visit to raw Exposure.

Table 24: Visit_To_RawExposure Table

Column Name	Data type	Unit	Description [UCD]
visitId	int		Pointer to entry in Visit table. [meta.id; obs.image]
rawExposureId	long		Pointer to entry in RawExposure table. [meta.id; obs.image]

7.25 prv_ProcHistory

This table produces unique procHistoryIds. The id changes each time something changes in the provenance. It is not linked to any other table. Because it is recording the time, it can

serve as a 'snapshot'. E.g., based on the time we can find out which configurations were valid at that time, what was executed at that time etc. It also serves as a 'flag' that something has changed.

Table 25: prv_ProcHistory Table

Column Name	Data type	Unit	Description [UCD]
procHistoryId	long		Unique id [meta.id; src]
theTime	timestamp		Time when this procHistory id was created.
description	text		Description what has changed. This is optional.

7.26 prv_Pipeline

This table defines all LSST Pipelines. One row per pipeline.

Table 26: prv_Pipeline Table

Column Name	Data type	Unit	Description [UCD]
pipelineId	int		Unique id [meta.id; src]
pipelineName	string		Pipeline name. [meta.id; src]

7.27 prv_cnf_Pipeline

This table defines all configurations for all pipelines.

Table 27: prv_cnf_Pipeline Table

Column Name	Data type	Unit	Description [UCD]
pipelineCnfId	int		Unique id [meta.id; src]
pipelineId	int		Id of the pipeline this configuration is for.
validityBegin	timestamp		Time when this configuration started to be valid.
validityEnd	timestamp		Time when this configuration stopped being valid.
notes	string		Notes/description useful to keep with this configuration.

7.28 prv_Task

This table defines all tasks for all pipelines.

Table 28: prv_Task Table

Column Name	Data type	Unit	Description [UCD]
taskId	int		Unique id [meta.id; src]
taskName	string		Name of the task.

7.29 prv_cnf_Pipeline_Tasks

This is a helper table for prv_cnf_Pipeline, it defines what tasks a given configuration of a pipeline consists of, and what the order is. If tasks are arranged hierarchically, parentTaskId determines which parent task given task belongs to. For tasks that do not have a parent task, parentTaskId is set to NULL. Positions should be numbered starting with 1, and are relative to its parent task.

Table 29: prv_cnf_Pipeline_Tasks Table

Column Name	Data type	Unit	Description [UCD]
pipelineCnfId	int		
taskId	int		Id of the corresponding task.
parentTaskId	int		Id of the parent task, or NULL if there is no parent task.
taskPosition	int		Position of the task in the pipeline. Starts with 1.

7.30 prv_cnf_Task

This table defines all configurations for all tasks. Note that occasionally manual patching will be required, leading to more than one configuration (the default one, and the patch). This is achieved through taskCnfVersion column.

Table 30: prv_cnf_Task Table

Column Name	Data type	Unit	Description [UCD]
taskCnfd	int		Unique id [meta.id; src]
taskId	int		Id of the corresponding task.
validityBegin	timestamp		Time when this configuration started to be valid.
validityEnd	timestamp		Time when this configuration stoped being valid.
taskCnfVersion	int		Version of the config (in case there is more than one configuration that is valid for a given validity range.
gitSHA	string		We need to capture version of the software used by this task. For now we are assuming it is just one SHA of one commit in git. In practice this can be more complicated, it can span multiple repos etc.

7.31 prv_cnf_Task_Columns

This table defines which tables+columns are altered by a given task. One row per table+column.

Table 31: prv_cnf_Task_Columns Table

Column Name	Data type	Unit	Description [UCD]
taskCnfd	int		If of the corresponding task configuration.
tcName	text		Table and column pair. Format: '<table>.<column>'. '<table>.*' is allowed to indicate all columns in a table.

7.32 prv_cnf_Task_Files

This table defines which files are altered by a given task. One row per file. This table can be trivially extended should we capture which sections of files are altered.

Table 32: prv_cnf_Task_Files Table

Column Name	Data type	Unit	Description [UCD]
taskCnfd	int		If of the corresponding task configuration.
fileUrl	text		url that uniquely locates the file.

7.33 prv_cnf_Task_KVParams

This table keeps parameter values for tasks. One row per parameter. For now everything is kept as strings (not efficient).

Table 33: prv_cnf_Task_KVParams Table

Column Name	Data type	Unit	Description [UCD]
taskCnfd	int		If of the corresponding task configuration.
theKey	string		Unique key.
theValue	string		Value for the given key.

7.34 prv_Node

This table defines nodes. One row per node.

Table 34: prv_Node Table

Column Name	Data type	Unit	Description [UCD]
nodeId	int		Unique id [meta.id; src]
nodeName	string		Name of the node.

7.35 prv_cnf_Node

This table defines all configurations for all nodes.

Table 35: prv_cnf_Node Table

Column Name	Data type	Unit	Description [UCD]
nodeId	int		Id of the node this configuration is for.
validityBegin	timestamp		Time when this configuration started to be valid.
validityEnd	timestamp		Time when this configuration stopped being valid.
ip	string		IP address (just a silly varchar for now)
os	string		Operating system name and version.
cores	int		Number of cores.
ram	int		Size of memory [GB].

7.36 prv_DataBlock

This table defines blocks of data. A block of data is a group of ids from the same table that are processed together using the same configuration.

Table 36: prv_DataBlock Table

Column Name	Data type	Unit	Description [UCD]
blockId	long		Unique id [meta.id; src]
tableName	string		Name of the table this data block is part of.

7.37 prv_RowIdToDataBlock

This table defines which rows belong to a given data block.

Table 37: prv_RowIdToDataBlock Table

Column Name	Data type	Unit	Description [UCD]
theid	long		The id of one data element. Note that we are not enforcing strict foreign key constraint because this will point to different tables.
blockId	long		Id of the block a given row id corresponds to.

7.38 prv_TaskExecution

This table keeps information about all tasks ever executed. Since the configuration of the system is not allowed to change while a task is running, we are not keeping a time range here, but instead we just keep the time of when the task started. It might be a good idea to keep the time of the middle of task execution: $\text{endTime} - \text{startTime} / 2$ to reduce changes of running into an issue with time synchronization between different machines. This table is one of the most important parts of provenance - it links task executions with nodes and processed groups. Occasionally manual patching will be required, which will lead to more than one valid configuration of a given task. TaskCnfVersion indicates which version should be used for a given task execution.

Table 38: prv_TaskExecution Table

Column Name	Data type	Unit	Description [UCD]
taskExecId	long		Unique id [meta.id; src]
taskId	int		Id of the task that is executed.
nodeId	int		Id of the node where the task is executed.
theTime	timestamp		The time when the task execution was started.
taskCnfVersion	int		Version of the task configuration. Typically there is only one, in some rare cases when manual patching is required, we can end up with more than one valid config version.

7.39 prv_TaskExecutionToInputDataBlock

This table maps task executions to input DataBlocks. Each block is typically processed by multiple task executions, and each task execution may process multiple data blocks, so it is many-to-many.

Table 39: prv_TaskExecutionToInputDataBlock Table

Column Name	Data type	Unit	Description [UCD]
taskExecId	long		Id of the task execution.

Continued on next page

Table 39: prv_TaskExecutionToInputDataBlock Table

Column Name	Data type	Unit	Description [UCD]
blockId	long		Id of the block a given task execution uses an input or NULL.

7.40 prv_TaskExecutionToOutputDataBlock

This table maps tasks executions to output DataBlocks. Note that each task execution may output multiple data blocks. There is one entry here for each task execution - output block pair.

Table 40: prv_TaskExecutionToOutputDataBlock Table

Column Name	Data type	Unit	Description [UCD]
taskExecId	long		Id of the task execution.
blockId	long		Id of the block produced by a given task execution or NULL.

7.41 sdqa_ImageStatus

Unique set of status names and their definitions, e.g. 'passed', 'failed', etc.

Table 41: sdqa_ImageStatus Table

Column Name	Data type	Unit	Description [UCD]
sdqa_imageStatusId	short		Primary key
statusName	string		One-word, camel-case, descriptive name of a possible image status (e.g., passedAuto, marginally-PassedManual, etc.)
definition	string		Detailed Definition of the image status

7.42 sdqa_Metric

Unique set of metric names and associated metadata (e.g., 'nDeadPix;', 'median;', etc.). There will be approximately 30 records total in this table.

Table 42: sdqa_Metric Table

Column Name	Data type	Unit	Description [UCD]
sdqa_metricId	short		Primary key.
metricName	string		One-word, camel-case, descriptive name of a possible metric (e.g., mSatPix, median, etc).
physicalUnits	string		Physical units of metric.
dataType	char		Flag indicating whether data type of the metric value is integer (0) or float (1).
definition	string		

7.43 sdqa_Rating_ForAmpVisit

Various SDQA ratings for a given amplifier image. There will approximately 30 of these records per image record.

Table 43: sdqa_Rating_ForAmpVisit Table

Column Name	Data type	Unit	Description [UCD]
sdqa_ratingId	long		Primary key. Auto-increment is used, we define a composite unique key, so potential duplicates will be captured.
sdqa_metricId	short		Pointer to sdqa_Metric.
sdqa_thresholdId	short		Pointer to sdqa_Threshold.
ampVisitId	long		Pointer to AmpVisit. [meta.id; obs.image]
metricValue	double		Value of this SDQA metric.
metricSigma	double		Uncertainty of the value of this metric.

7.44 sdqa_Rating_CcdVisit

Various SDQA ratings for a given CcdVisit.

Table 44: sdqa_Rating_CcdVisit Table

Column Name	Data type	Unit	Description [UCD]
sdqa_ratingId	long		Primary key. Auto-increment is used, we define a composite unique key, so potential duplicates will be captured.
sdqa_metricId	short		Pointer to sdqa_Metric.
sdqa_thresholdId	short		Pointer to sdqa_Threshold.
ccdVisitId	long		Pointer to CcdVisit. [meta.id; obs.image]
metricValue	double		Value of this SDQA metric.
metricSigma	double		Uncertainty of the value of this metric.

7.45 sdqa_Threshold

Version-controlled metric thresholds. Total number of these records is approximately equal to 30 x the number of times the thresholds will be changed over the entire period of LSST operations (of order of 100), with most of the changes occurring in the first year of operations.

Table 45: sdqa_Threshold Table

Column Name	Data type	Unit	Description [UCD]
sdqa_thresholdId	short		Primary key.
sdqa_metricId	short		Pointer to sdqa_Metric table.
upperThreshold	double		Threshold for which a metric value is tested to be greater than.
lowerThreshold	double		Threshold for which a metric value is tested to be less than.
createdDate	timestamp		Database timestamp when the record is inserted.

7.46 LeapSeconds

Based on <http://maia.usno.navy.mil/ser7/tai-utc.dat>.

Table 46: LeapSeconds Table

Column Name	Data type	Unit	Description [UCD]
whenJd	float	d	JD of change in TAI-UTC difference (leap second). [time.epoch]
offset	float	s	New number of leap seconds. [time.interval]
mjdRef	float	d	Reference MJD for drift (prior to 1972-Jan-1). [time.epoch]
drift	float	s/d	Drift in seconds per day (prior to 1972-Jan-1). [arith.rate]
whenMjdUtc	float	d	MJD in UTC system of change (computed). [time.epoch]
whenUtc	long	ns	Nanoseconds from epoch in UTC system of change (computed). [time]
whenTai	long	ns	Nanoseconds from epoch in TAI system of change (computed). [time]

7.47 ApertureBins

Definition of aperture bins (for both the Object and Source tables.)

Table 47: ApertureBins Table

Column Name	Data type	Unit	Description [UCD]
binN	byte		A bin in radius at which the aperture measurement is being performed.
radiusMin	float		Minimum aperture radii of bin.
radiusMax	float		Maximum aperture radii of bin.

References

- [1] **[LSE-163]**, Jurić, M., et al., 2017, *LSST Data Products Definition Document*, LSE-163, URL <https://ls.st/LSE-163>

Draft