VISTA DATA FLOW SYSTEM (VDFS)

for VISTA & WFCAM data

Science Archive Database Design Document

author
N.J.G.Cross (WFAU Edinburgh)
Science Archive Project Scientist
number
VDF-WFA-VSA-007
issue
Issue 1.0
date
30 Sep 2006
co-authors
N.C.Hambly, R.G.Mann, E.T.W.Sutorius, M.A.Read

Contents

1 SCOPE

The Science Archive database design document describes how pixel, catalogue and other data are stored within the VSA and WSA science archives. The document is intended to be a technical reference for archive implementation, curation and also for development of the user interface layer. The design is expressed in sufficient detail to enable implementation of the VSA system, to enable development of the required curation software, and to inform development of the user interface software.

The external requirements on the VSA design are expressed in the following applicable documents (which are listed in Section 11): science requirements and their consequent functional requirements are detailed in the VSA Science Requirements Analysis Document (SRAD; AD01). The SRAD itself follows on directly from the outline survey designs that form the basis of the UKIRT Infrared Deep Sky Survey proposal (AD02) which form the core usage of the VSA. The characteristics of the ingested data are detailed in the CASU/WFAU Interface Control Document (ICD; AD03). Top–level dataflows are described in the VSA Data Flow Document (DFD; AD04). Relevant details of the archive hardware design can be found in the Hardware/OS/DBMS Design Document (HDD; AD05). Likely modes of science exploitation of the WSA are illustrated in Usages of the WSA (AD06). Details of the management and planning can be found in the VSA Management and Planning Document (MP; AD07). Finally, the software architecture of the VSA and WSA are described in the VSA Softwate Architecture Design document (AD08).

The relational design of the archive follows on directly from the functional requirements expressed in the SRAD, and curation 'use cases' (see Section 4). Relational models are presented that show the data entity (table) organisation along with their associated attributes (fields). These entity–relationship models are mapped onto tables, and the database schemas are described. Processing requirements are developed from the curation use cases and details of the additional processing for curation are given.

2 OVERVIEW

This document is structured as follows. In Section 3, we discuss the fundamental concepts behind the database design, including pragmatic choice of DBMS, overall archive organisation, and calibration considerations. In Section 4 we describe a set of curation 'use cases' which have been developed along with the SRAD functional requirements to enable database design. In Section 5 we introduce the relational design of the databases that will comprise the VSA, including entity–relationship models (ERMs) that provide both general, illustrative overviews of the database structure and specific detail of the VSA data. Section 6 goes on to develop the relational design of the VSA archive into implementation of prototype database tables within the chosen DBMS; details of tables, column names, unique identifiers and primary and foreign keys are given. We also describe logical table 'views' that will be created. In Section 7 we discuss issues of indexing within the database, give details of logical entities/attributes for enhanced utility, and describe defined functions that will be available for both curation and use. Finally, in Section 8 we give details of the application software that is being developed for the purposes of data transfer, ingest and curation.

3 FUNDAMENTALS

3.1 Relational versus object–oriented design

VSA design will be based on a relational DBMS. Throughout the planning phases of the VSA/WSA, we have followed closely the development of the science archive system for the Sloan Digital Sky Survey (SDSS) at Johns Hopkins University [2]. The pragmatic approach for VSA design has always been to use, wherever possible, the same or similar archive systems as SDSS since the latter is state–of–the–art in astronomical archives and is based on many staff years of effort that realistically are not available to WFAU. Originally, the SDSS science archive was based on the OODBMS Objectivity. However, since the Early Data Release (EDR) performance issues have arisen (e.g. concerning indexing [6]) and the subsequent SDSS releases (DR1-5) have been made under an RDBMS system known as SkyServer (it should perhaps be noted that a relational design results in an archive system that is more intuitively understandable to the non–expert user).

3.2 RDBMS choice

The VSA has been implemented using the Microsoft DBMS 'SQL Server' running under the Windows 2003 'Advanced Server' operating system. The SDSS releases in 'SkyServer' use SQL Server, and once again we are benefitting from the SDSS experience and developments in following this choice. Note however that we are not tied to SQL Server for future developments. If this choice proves not to be scalable to data volumes expected to accumulate after several years of operation, then we will use one of the other two large commercial RDBMS products currently available: IBM's DB2 or Oracle (additional information is available in AD05). It should be noted that the interface to all these DBMS products is industry–standard SQL, so application scripts and code are not wedded to SQL Server (see the software architecture document, AD09).

3.3 Arrangement of data

Pixel data will be stored as flat files rather than 'binary large objects' (BLOBs) internal within the DBMS. This is so that high data volume image usages that are not time critical will not impact datamining usages on catalogues where more 'real time' performance is required for data exploration/interaction. However, pixel files and the pixel metadata will be tracked in tables within the DBMS so that the image metadata can be browsed/queried in the same way as any other data (catalogues and ancillary data will of course be stored in tables within the DBMS).

3.4 Null values

All attributes in the VSA will be 'not null'. Inevitably, from time to time attribute values will be unavailable or not applicable in certain situations. RDBMSs provide facilities to handle this eventuality, with nullable attributes handled by the system. However, making use of this facility can complicate curation and use enormously (e.g. SQL can become cumbersome). On advice from Jim Gray of Microsoft Research, and following the SkyServer example and our own experience so far, we assert that all attributes must have externally specified values and when not available or not applicable, a specific meaningful default value will be inserted. Where null values are referenced to other tables via foreign keys, a null record will be included in the referenced table (this record consisting, in turn, of externally specified null values for all attributes). Table 1 lists the default null values in the science archive.

3.5 Curation information

The VSA databases will contain tables giving details of curation information, for the purposes of curation housekeeping and also for informing users of updates and processing that have taken place. Further, the VSA will contain tables of constants and other generic information (e.g. a filter table giving details of the filter attributes, a programme table giving details of the surveys), again for housekeeping and informing users.

Table 1: Default values for various data types in the science archive database.

Default value	Data type
-0.9999995×10^9	Floating point (single/double precision)
−99999999	Integer (4– and 8–byte)
−9999	Integer (2–byte)
NONE	Character
9999–Dec–31	Date–times

3.6 Calibration

The VSA will store astrometrically and photometrically calibrated quantities via attributes stored in calibration–independent units along with calibration coefficients, but will also store calibrated quantities based on the most recent calibration available for efficiency and ease of use. It is a specific requirement (see the SRAD) that recalibration of photometric and/or astrometric quantities must be possible within the VSA, and further that previous calibrations be available for use in addition to any current (and presumably 'best', or at least up–to–date) version. For instance the previous zeropoint for each image will be available along with a timestamp, and the previous zeropoint for each detector will also be available along with a timestamp. Previous and current values of the photometric solution (see § 5.4.3) will also be found in the archive.

3.6.1 Astrometry

Spherical co-ordinates will be stored internally within the science archive in Equinox J2000.0 celestial RA and Dec tied to the ICRS via secondary astrometric standards (2MASS). Astrometric calibrations will be specified via zenith–polynomial or tangent plane projections – see, for example, the ICD (AD03) and references therein – for non–resampled and mosaiced/resampled images respectively. Hence, astrometric calibration coefficients will consist of zeropoints and the 'CD' linear coefficients (for tangent plane projections) and additionally 'PV' non–linear radial distortion coefficients for zenith projections). Calibration–independent XY co-ordinates will always be associated, via relationships to images, to their calibrating coefficients. Low–level, highly non–linear distortions that may become apparent after several years of accumulation of data (analogous to the 'swirl' distortion maps present on Schmidt plates – e.g. [7]) will be allowed for in the calibration entities and calibration scheme. Finally, spatial indexing attributes will be allowed for every occurence of celestial co-ordinates in the archive to enable efficient curation and usage employing positions (e.g. source merging, proximity searching).

3.6.2 Photometry

The photometric calibration of WFCAM data is mainly done during CASU processing by comparison to 2MASS and this is described in detail in Hodgkin et al. (2006). Over time improvements to the calibration will naturally occur, especially for the bluer Z and Y filters, where extrapolation from 2MASS is necessary, and also the narrow band filters, e.g. the J_{118} filter in VIRCAM. In the future additional improvements may be made using a photometric solution such as that described in the Science Archive Photometric Recalibration Design Document [9].

3.7 Reruns, repeats and duplicates

Multiple instances of a given image or source detection will inevitably occur within the VSA. There are several reasons for this:

- pixel data may be reprocessed and/or rerun through altered source extraction procedures, particularly in the early stages of commissioning of the data flow system. Newly processed image data and corresponding source detection lists, re-ingested, will result in repeats of the same data that must be uniquely identified and flagged as such.
- survey design will result in overlap areas between adjacent fields, resulting in duplicate detections of the same objects at different times – useful data that will be retained in the archive but that must be distinguishable for the purposes of duplicate–free source lists. At present the source that is selected is the one with the most pass band information. e.g. if there are two sources and one has been observed in Y,J,H & K and the other in H & K only, then the former is chosen.
- stacked survey products will be recreated after some observing period, resulting in ingestion of updated image products and higher s/n detections of previously ingested sources – again, these must be distinguishable when retained in catalogues.

Where necessary, all records will be flagged as deprecated where repeats are potentially present, and all potentially duplicated records will have a primary/secondary attribute specified for one primary, all other records being identified as secondary. The intention is to retain old versions of processed image products and their source detections so that users can refer back to observations previously used to make target lists for follow–up programmes. Of course, retention of large amounts of pixel data has to be subject to storage constraints – the 'best' data (i.e. most recently processed) must take precedence when disk space is limited.

In cases where data is recalibrated, the image or catalogue data will not be replicated, but the old calibration coefficients (e.g. magnitude zeropoints) will be logged and timestamped so that older versions of the catalogues can be reproduced easily.

4 Curation use cases

A set of curation 'use cases' expressed in plain language are reproduced from [16] in Appendix 9.1. The key points to emerge from this exercise with respect to the archive relational design are:

- updates to the database occur on a variety of timescales, from daily ingest to quarterly (or possibly less frequent) 'releases';
- housekeeping (curation) entities emerge as being necessary to keep track of updates as survey data accumulate;
- new attributes and relationships between observational entities emerge that are required for the purposes of curation.

In the next Section, these points are propagated into the database design. The curation use cases are expanded into activity flow diagrams in Section 8 for the purposes of detailing the curation software design.

5 Science Archive relational models

5.1 Archive databases

There are several fundamental requirements that come directly from the SRAD and VSA usages (AD01 and AD06) and curation use cases:

- ingest occurs on a daily basis, and a complex set of curation tasks must be undertaken on a periodic basis to produce useful survey catalogues – hence, the concept of *offline* and *released* databases emerges;
- data having various proprietory periods will be stored within the archive and proprietorial rights must be enforced;
- survey programmes consisting of different combinations of passbands within given fields, and different observing modes, will be ingested and stored;
- externally produced survey datasets will be held in the archive;
- browsing of the *descriptive* data (i.e. details of what is held in the archive) will be unrestricted, even when access to the data themselves is restricted.

For example, it might be considered possible to have a small number of database tables in a single database to simplify curation. However, on examination this scenario will not work – e.g. a merged source catalogue becomes excessively complicated since at any given time some data may be open to all users while other data may be still be subject to proprietorial rights. These considerations result in a top–level archive organisation that is illustrated in Figure 1. Each box within the overall VSA illustrates a different physical database that will be stored and curated. The key points to note are:

- the 'WFAU database' is the offline, daily/periodically updated version of all archived data, these being copied out to a publicly accessible catalogue server at release time;
- within the database, the observations that are stored form natural groups with different characteristics; these groups are identified as 'entities' and their characteristics as attributes in the relational design detailed in the next Section.

5.2 A brief introduction to entity–relationship modelling

Entity–relationship models (ERMs) have been employed in the design of the VSA. Simply put, an *entity* is something of significance to the system data that appears many times with a fixed set of *attributes*; relationships (generally one–to–many) naturally occur between entities. The advantages of ER modelling are:

- it provides a simple, pictorial summary of the relational design for the database;
- ER modelling is RDBMS–independent, and results in a design that can be easily mapped into any RDBMS;
- ERMs map directly into a table design: entities become tables, attributes become columns and relationships become *foreign key constraints* between the tables.

Here we will use ER modelling in its simplest form to described the VSA relational design. To illustrate the technique, an example ERM is shown in Figure 2. This is for the SuperCOSMOS Sky Survey [17] (SSS), hereafter known as the SuperCOSMOS Science Archive (SSA); the SSA was used to prototype many aspects of the V1.0 WSA hardware and interfaces. In the diagram,

- entities are shown as boxes with rounded corners ('softboxes');
- a few attributes for each entity are listed – a unique attribute is labelled with # while * denotes any other mandatory (but not necessarily unique) attribute;

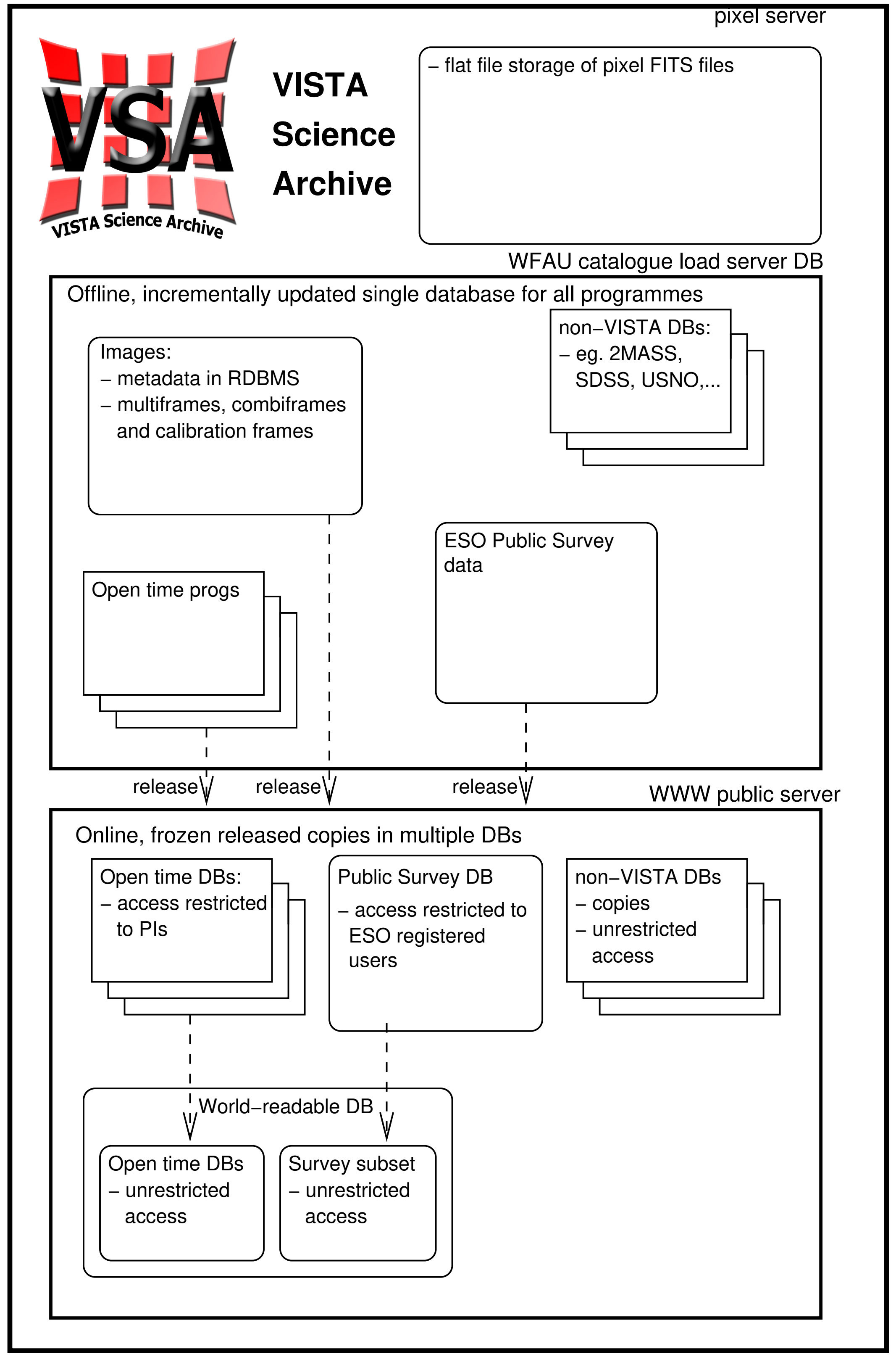

Figure 1: *Schematic overview of the VSA. Within the archive, there are three distinct data servers: pixel flat files, offline DBMS and online DBMS. The online DBMS contains several distinct databases.*

- relationships are illustrated with a single line forking out for a one–to–many relationship, dashed lines illustrate optional relationships;
- a barred relationship is used as a shorthand way of noting propagation of a unique attribute from one entity to another.

So, in Figure 2 for the SSA, we have entities Survey, Plate, Field, Detection and Source where

- every merged Source consists of one or more survey Detections;
- every survey Detection belongs to one of the survey subtypes B, R1, R2 or I;
- every plate yields one or more survey Detections and belongs to one survey and one field;
- every field is covered by one or more plates (actually 4 in the SSA);
- every survey consists of many plates in many fields.
- barred relationships indicate combinatorial unique identifiers (UIDs) in entities where the specified attribute is not unique – in the case of the SSA, field numbers in entity field, the combination of the survey UID plus the field ID results in a UID for a field (e.g. field 319 exists in the ESO/SRC field system in the southern hemisphere and also in the POSS–I and POSS–II systems in the northern hemisphere, all three at different co-ordinates).

The resulting schema file for the SSA, as implemented in SQL Server both as a standalone prototype and as one of the external surveys required to be stored for cross–identifications within the WSA and the VSA, is available online SuperCOSMOS Science Archive

In the following subsections, we provide and discuss the ERMs for the VSA. For clarity, these are divided into top–level data, pixel data, calibration data, catalogue data, external data and curation/housekeeping data. The final subsection discusses how these ERMs fit into the databases in the overall archive as described previously.

5.3 Pixel data

Image files are stored within the VSA & WSA as Multiframes. These are linked to the flat multiextention FITS (MEF) files. Multiframes contain the MEF filename, image type, filter, observation information (e.g. pointing, guide star, observation date), grouping information (e.g. number of jitters, microsteps, dithers and positions of each), zeropoint, telescope imformation (e.g. humidity, mirror temperature) and the data reduction information (e.g. IDs of the dark, sky and flat field imformation. Most of this information is the metadata stored in the primary header of the MEF file.

Each MEF file also contains one or more extensions (one in the case of a *tile* or *mosaic*; 16 in the case of a *normal* observation — a reduced science frame). The detector information consists of world coordinate information and metadata related to the catalogues, such as seeing and aperture corrections. This is stored in the MultiframeDetector entity.

The grouping of multiframes is one area where there are significant differences between the VSA and the WSA. In the WSA a typical group of images would contain 8 *normal* images, each with 4 extensions (corresponding to the 2×2 array of detectors). These are split into two smaller groups of 4 frames; each group is separated by a small jitter offset (typically a few pixels). The four frames in each small group is taken in an 4-point microstepped pattern, so that a higher resolution image can be made by interleaving these frames. Each extension in a normal image is 2048×2048 pixels. Each group of *normal* images is interleaved to form one *interleaved* image, also composed of 4 extensions, but each

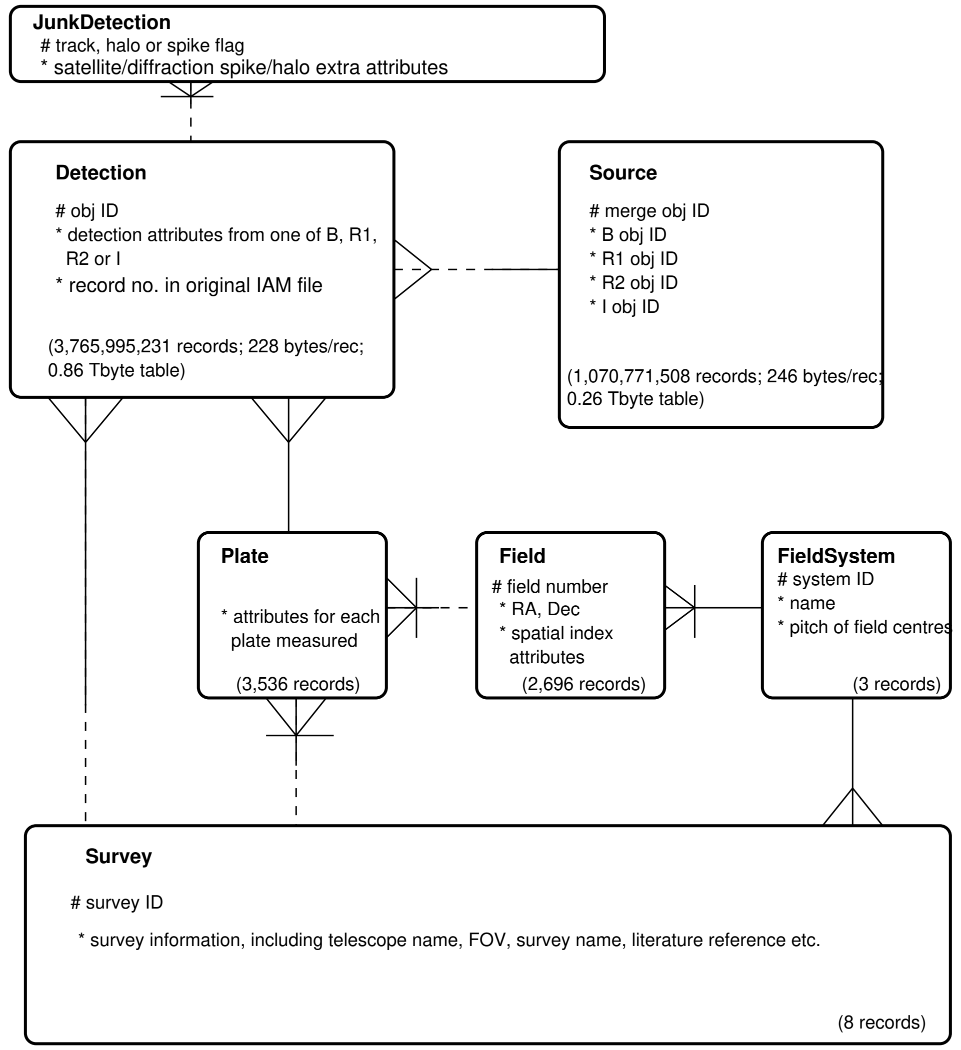

Figure 2: *Entity–relationship model for the SuperCOSMOS Science Archive, an implementation of the SSS [17] in a commercial RDBMS (relative sizes of the different entities are indicated for information only – this is not standard ERM practice).*

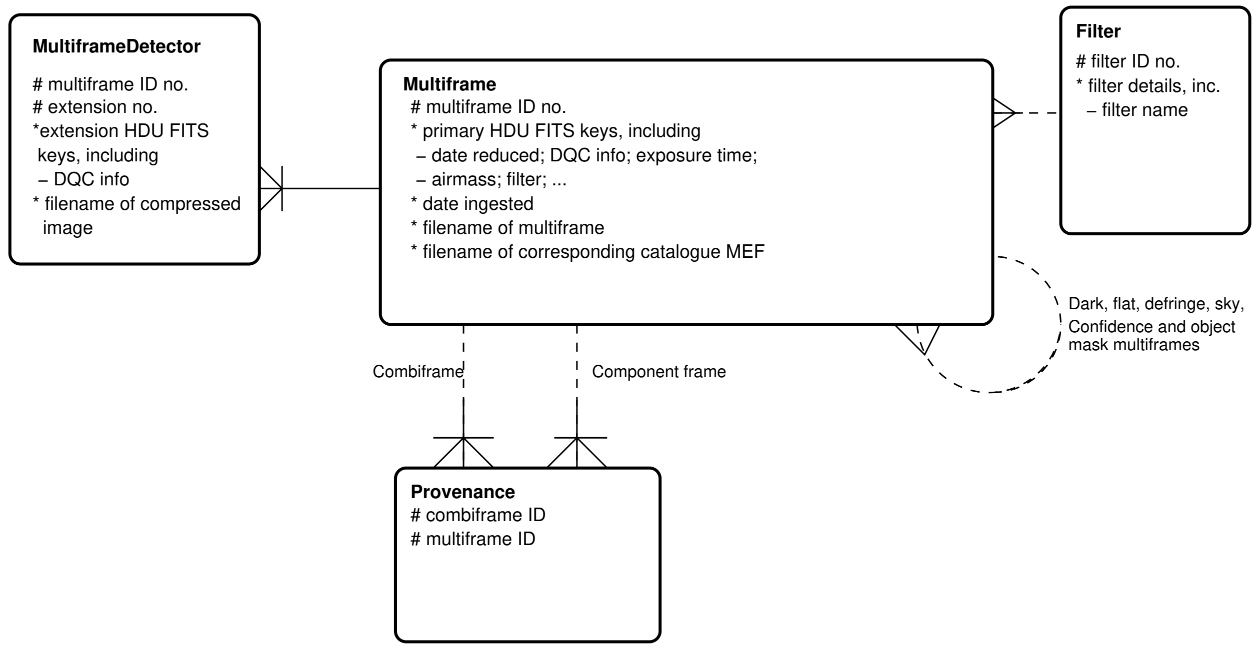

Figure 3: *Entity–relationship model for image data in the VSA. Each box represents a table in the database; the lists of attributes in each are for illustration only and are not intended to be complete.*

extension containing $\sim 4k \times 4k$ pixels. The 2 *interleaved* images are then stacked to form one *leavstack* image, of similar dimensions. Most stack frames which are used to generate catalogues in the WSA are pawprints with 4 extensions, each with $\sim 4k \times 4k$ pixels. Mosaicing of these stacks is rare.

In the VSA interleaving will not happen as much, since the pixel size of VIRCAM is better matched to the typical seeing. However, unlike WFCAM most normal frames will be mosaiced to form a *tile* with a single extension. A typical VISTA group may contain 6 *normal* images, each with 16 extensions. Rather than being microstepped to improve the resolution, these are jittered in a six-point pattern — 2 in the x-direction, 3 in the y-direction — to fill in the tile. Each tile will be $16k \times 13k$, with most parts observed twice. In the WSA the few mosaics produced are mosaics of deep stacks. In the VSA deep stacks will be stacks of tiles. The relational model for image data is shown in Figure 3. Key points to note are:

- many science frames in Multiframe are associated with other science multiframes with one being a *normal* frame that is part of a *tile* frame or a *tile* frame that is part of a *tilestack* frame. These relationships are mapped using the Provenance table
- every Multiframe is associated with one filter, and consists of one or more detector frames in MultiframeDetector (usually 16 for a VIRCAM pawprint);
- many frames in Multiframe have associated library calibration multiframes *Dark*, *Confidence*, *Flatfield* and *Sky*, which in turn have detector frames. These are represented the the dashed loop connecting multiframe with itself.

This same model was used in the WSA. There can be several layers of interleaving, stacking and mosaicing, so many frames can be listed in both the combiframeID column and the multiframeID column of Provenance. In the UKIDSS-UDS for instance there are mosaic images where thousands (tens of thousand when it is complete) *normal* frames are interleaved in a 3×3 microstep pattern. These are then stacked within each observing block. Each of these stacks are then stacked again to create a deep stack in each pointing. Finally each of these pointings are mosaiced together to create a single image for each filter.

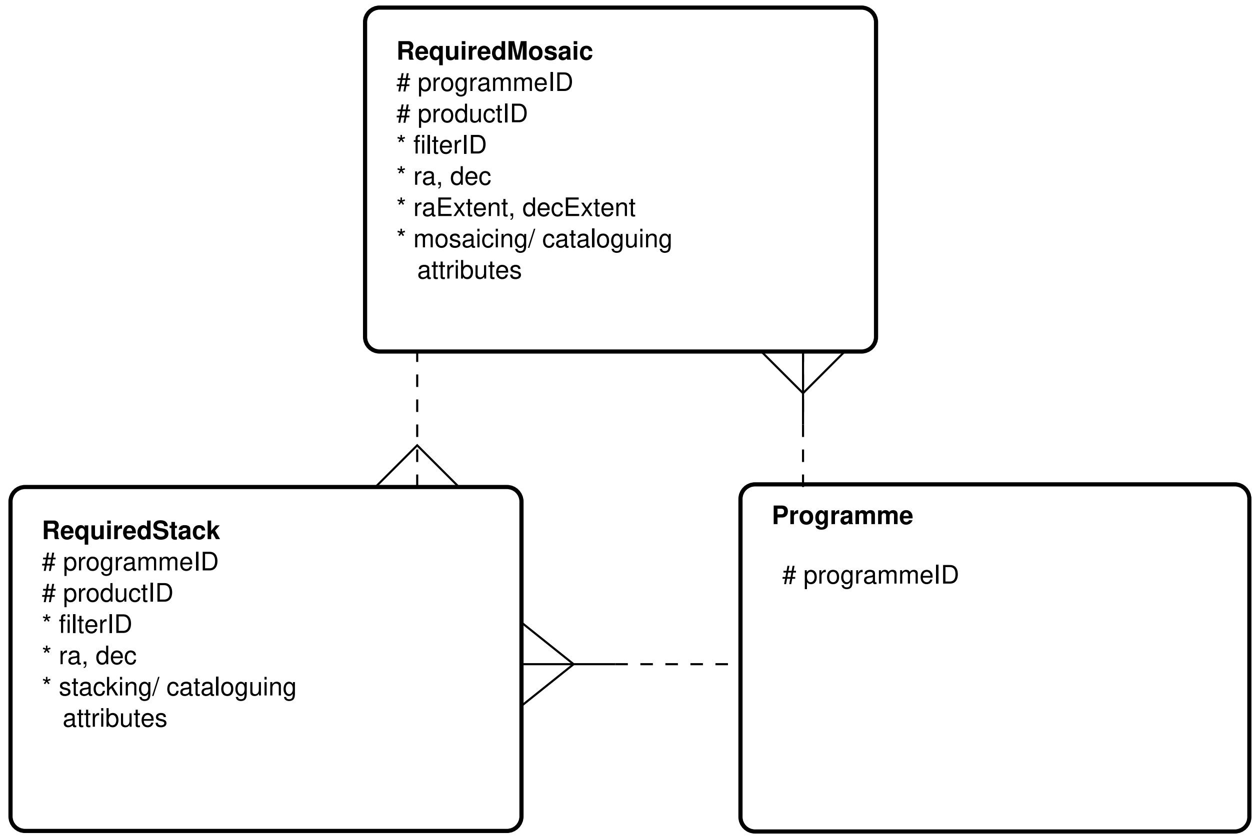

Figure 4: *Entity–relationship model for required stacks and mosaics in the VSA. Each deep stack or mosaic is specified beforehand in the RequiredStack or RequiredMosaic table.*

In the image ERM, filename attributes associate all the metadata with i) the image FITS file ii) for science frames, the catalogue file ingested from CASU with the image or generated using the same standard CASU source detection software; iii) the filename of a compressed image for speedy quick–look finder charts and or browsing. All these files will be stored externally to the DBMS in the mass–storage flat file Linux file system (AD05).

5.3.1 Deep Stacks and Mosaics

Deep stack and mosaic multiframes are stored in the VSA in the same way as multiframes from shallow surveys. In addition to this, however, the stacks and mosaics are defined beforehand in RequiredStack and RequiredMosaic. The ERM for required stacks and mosaics is shown in Figure 4. Stacks are built up in pawprints and then these pawprints are joined together to make the mosaic. For each mosiac in the programme, RequiredMosaic defines the filter, the centre, and the size of the mosaic. It also defines whether the mosaic is to be rebinned to a larger pixel size and which software to use to mosaic. The size (raExtent and decExtent), the centre and the filter arguments from RequiredMosaic are used to select the stacks from RequiredStack.

RequiredStack lists requirements for stacking including the data quality control requirements. However, many programmes that require deep stacks do not require mosaics, so RequiredStack is also directly linked to Programme. Both RequiredStack and RequiredMosaic give the catalogue extraction requirements: the extraction tool to use, and values for thresholding and background estimation.

5.3.2 Quality Assurance of pixel data

The design of the VSA/WSA includes provision of features to enable general quality control (QC) of ingested data. Such features as a deprecation code attribute in every table subject to ingest modification, and expurgation of deprecated data in final released database products are provided. General QC is necessarily a rather open–ended problem requiring much interaction with the data, at least in these initial stages of survey operations. Although the science archive design does not preclude fully automated QC procedures, presently the WSA UKIDSS data (for example) have a lengthy semi–automated QC process applied, some details of which are given in [10] and [11]. Table 2 provides details of the QC checks applied to UKIDSS data as they stand at the time of writing. Note, however, that for UKIDSS released database products all deprecated data are removed, so users will see only those data records having attribute `deprecated`=0. Presently, no QC procedures are applied to non–survey data held in the WSA.

For VSA PSPs, the QC procedures will be based on the UKIDSS procedures, but will be adapted as required. For instance new observing procedures at UKIRT have eliminated the moon ghosts that were seen in early WFCAM frames and significantly reduced persistance. Learning from this will not only improve future WFCAM data, but also VIRCAM data.

5.4 VIRCAM catalogue data

5.4.1 Generalised ERM for catalogues within the science archive

Figure 5 illustrates a generalised ERM for VISTA catalogues that is applicable to all archived programmes. The following points are worthy of note:

- CASU standard processing results in a set of detections on every stacked science image (a single exposure or undithered set of interleaved images is typically not catalogued). Every detection is associated with an image and is uniquely identified via a sequence number within that image's detection list, and by the UID of the image. In this way, source detections are intimately linked to their progenitor image and all associated metadata. Both uncalibrated and calibrated instrumental quantities will be stored in the detection list – photometric fluxes and magnitudes; a *deprecated* flag attribute will distinguish repeat measurements (e.g. from re-ingested frames or updated stacks etc.);

- a merged source entity will be created using a merging procedure (one of 20 curation tasks – see Section 8) to create a single multi–colour, multi–epoch record to the prescription available for a given programme (these prescriptions are stored as one of several curation entities – see Section 5.6). The merged source list will include a primary/secondary attribute to identify duplicates (e.g. from overlap regions between survey fields). Enhanced, calibrated astrometric solution attributes will be included based on all available position measures at individual epochs. Additional attributes for a full–blown astrometric solution will also be included: α_0, δ_0, σ_{α_0}, σ_{δ_0}, μ_α, μ_δ and errors, epoch, χ^2 of astrometric fit, and number of position measurements. Calibrated photometric attributes will be included along with the subset of the 80 standard detection attributes (see ICD Section 6.2.2) considered most useful for science exploitation, i.e. including a subset of the flux/flux error measurements (as magnitudes), the basic profile attributes (Gaussian sigma, ellipticity and position angle), classification attributes and the original sequence number, leaving 23 attributes per passband. Calibrated fluxes will be stored in the merged source list, the calibration being based on the most recently derived photometric calibration; colours (e.g. J–K) will also explicitly stored as columns. For programmes where a significant number of the objects are expected to be extragalactic, the galactic extinction corrections in every band will be calculated. These are not calculated for programmes such as the UKIDSS Galactic Plane Survey

Table 2: WSA quality control deprecation codes and their meaning.

Deprecation code	Description
1	Stack frames that have no catalogue
2	Dead detector frames or all channels bad
3	Undefined and or non-sensible critical image metadata attributes
4	Sky subtraction not OK (via pipeline sky sub scale factor)
5	Incorrect combination of expTime,numExp,numInts for survey specific projects
6	Incorrect frame complements within groups/nights (for incomplete MSBs)
7	Undefined values of critical catalogue attributes for stacks
8	Seeing=0.0 for a stack
9	High value of sky that compromises the depth
10	Seeing outside specified maximum
11	Photometric zeropoint too bright
12	Average stellar ellipticity too high
13	Depth (as calculated from sky noise and 5sigma detection in fixed aperture) is too shallow compared to overall histogram distribution (i.e. shallower than 0.5mag wrt the modal value) OR sky noise is too high given the sky level
14	AperCor3 outlying in aperCor3 versus seeing distribution
15	Pipeline MAGZPT inconsistent between image PHDU, extension HDUs and/or catalogue extension HDUs (from attributes photZP, photZPExt and photZPCat)
16	Difference in detector sky level wrt to mean of all 4 detectors is outlying in the distribution of the same
18	Provenance indicates that a constituent frame of a combined frame product includes a deprecated frame
19	Inconsistent provenance for a stack or leav product indicating something wrong with the image product (usually corrupted up FITS keywords confusing the pipeline)
20	Detector number counts indicate some problem (loads of crud images)
21	5-sigma depth of detector frame more than 0.4mag brighter than modal value for a given filter/project/exposure time
22	Basic astrometry check (pixel size and/or aspect ratio) indicates something is wrong with the image
26	Deprecated because frame is flagged as ignored in pipeline processing
40	Science (*stack) frame is not part of a survey (e.g. high latitude sky frames in the GPS)
60	Eyeball check deprecation: trailed
61	Eyeball check deprecation: multiple bad channels
62	Eyeball check deprecation: Moon ghost
63	Eyeball check deprecation: Sky subtraction problem
64	Eyeball check deprecation: Disaster (catchall category for the indescribable)
65	Eyeball check deprecation: Empty detector frame
66	Flat fielding problem
70	Eyeball check requires deprecation, but this is the best that can be done so this should not be reobserved (e.g. very bright star in WFCAM FOV)
80	Deprecated because observation (MSB,object,filter) has been repeated later (shallow surveys only). The latest duplication in each case is kept
99	Manually deprecated because of some DFS issue (e.g. pipeline malfunction)
100	Multiframe deprecated because all detectors have been previously deprecated (and the MF not already deprecated)
101	MultiframeDetectors deprecated because parent Multiframe is deprecated (and the MFD not already deprecated)
102	Detection deprecated because parent MFD deprecated
> 127	Frame deprecated because reprocessing supersedes it

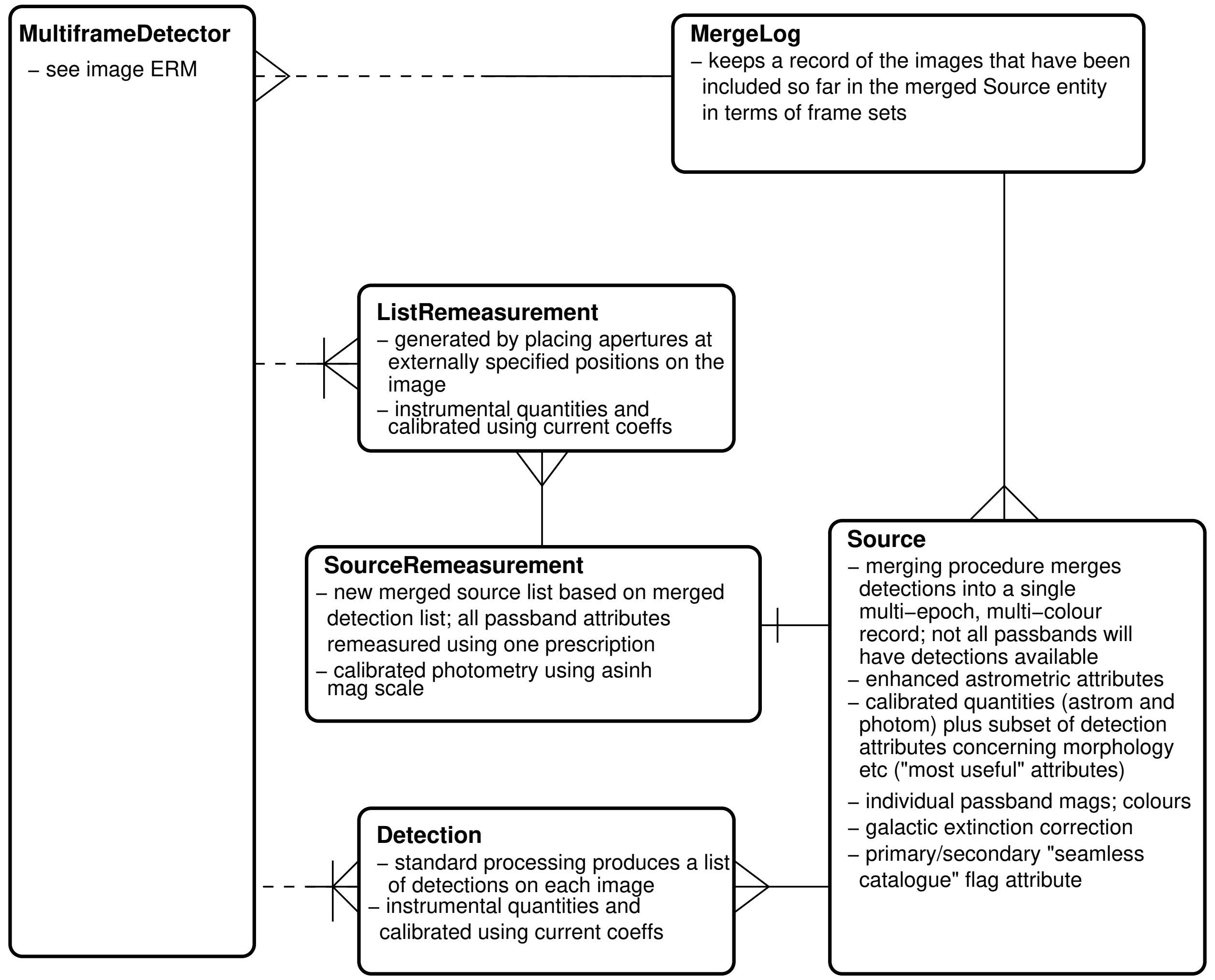

Figure 5: *Generalised ERM for VISTA catalogue data of all archived programmes.*

in the WSA because a large amount of CPU time would be wasted calculating an attribute that is meaningless for stars within our galaxy, which are seen at various distances within the dust layers, and highly uncertain (in areas of extremely high extinction) close to the galactic centre.

- for every archive programme, a merge log will be kept in linking the multiframes of each bandpass to the source catalogue. The super set of different bandpass science frames in any one pointing is called the *frameSet*. This log will have attributes frameSetID, ra, dec, and multiframeID and extNum for each image whose detections have been merged into the merged source list along with a deprecated flag (for the subsequent list–driven re-measurement curation task – see later). The final attribute will be a merge software version, to keep track of any necessary re-merging when the software has been updated.

- the other two entities in this generalised model contain list re-measurements for a particular programme's frame set. The idea here is that the merged source list will contain gaps (i.e. some passbands will not have detections corresponding to those in other passbands), and furthermore merged detections will be based on attributes produced from analysis of individual passband data in isolation. The concept of list–driven source analysis is to take the merged source list, and to define a master list of positions, apertures, profiles and deblending criteria and then apply them uniformly to all available passbands. This procedure results in a set of list–driven re-measurements in each passband which will be written to a *list re-measurements* entity analogous to the detection entity (and therefore relating back to the source image in a similar way) and with the set of attributes and with fluxes that *may be negative* (due to random errors). Similarly, the list–driven re-measurements, which are generated across all appropriate passbands in one go, will be written into a source catalogue analogous to the merged source detection entity. The relationship between the merged source detection entity and list–driven remeasurement source entity is one–to–one since every merged source detection will give rise to a list remeasurement across the passband set for a particular programme.

- for programmes which require deep stacks or mosaics, the release versions of Source and Detection entities contain data only from catalogues extracted from deep stacks and mosaics. For instance the UKIDSS DR1 release [11] contains UDS data from the deep J mosaic and the deep K mosaic only.

5.4.2 ERMs for synoptic data in the Science Archive

Synoptic data is data in the same pointing observed many times in the same filter to measure time varying characteristics, mainly changes in brightness or colour, but also position on the sky. These include intrinsically variable stars, such as RR Lyrae and Cepheid variables and outburts stars; eclipsing binary stars or stars with transiting planets; solar system objects; high proper motion stars; AGN; supernovae in other galaxies.

Various programmes within the WSA/VSA require synoptic data. The science archive must be able to handle this data in away which is logical, simple to use and fast to access. Figure 6 illustrates an ERM for synoptic data in a single band. The following points are worthy of note:

- the data flow consisting of the MultiframeDetector, Detection and DetectionNeighbours entities is currently in place for the UKIDSS calibration data and the Planetary Transit Survey in the WSA. This method is simple, but it produces an enormous calDetectionNeighbours table (currently 6.4 billion rows for 50 million detections in calDetection). The calDetectionNeighbours table is so big because each detection of an object in any pointing is crossed with every other detection of itself and nearby neighbours. For instance if a sparsely crowded region is observed 50 times in 5 filters then there are 250 unique detections which are crossed with each other

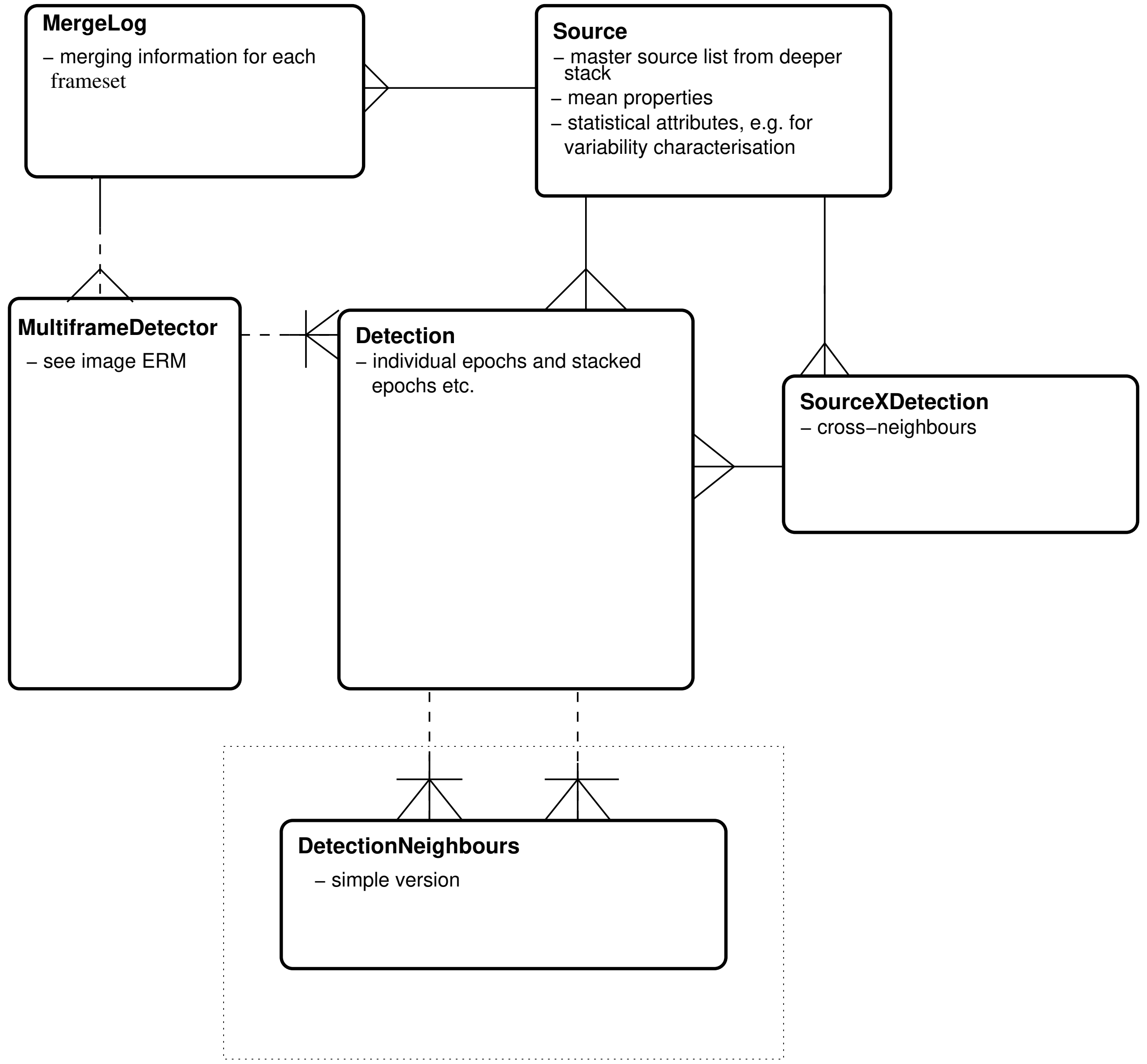

Figure 6: *ERM for synoptic data in VSA/WSA programmes.*

giving 250^2 rows in calDetectionNeighbours, where 50 would be useful. In a crowded region, this problem is magnified as nearby neighbours are also crossed with the main source. When the synoptic source merging code has been developed and tested, the DetectionNeighbours tables will become obsolete, and this is shown by the dotted box.

- the rest of the ERM in Figure 6 shows the improved synoptic pipeline, which is still being built. A subset of the best individual observations in each filter will be stacked and extracted to produce deep detection lists in the Detection table § 5.3.1. Each bandpass will be merged to form a Source table which will be the master list from which everything else is generated. This source list is then crossed with the detection table to form a SourceXDetection table. This table will match each detection of a particular object to the source table which contains a single row for it. This cuts down the number of rows in the above example neighbour table to 250, which is much simpler. Using the SourceXDetection table, it is easy to produce a light curve for any particular source in the Source table. This is the best solution for programmes which are synoptic in one band (e.g. the PTS or the VPSP VVV) or where data in each band is taken at different intervals to the other bands. This method takes advantage of much of the source code that has already been written for stacking and extracting deep data § 5.3.1, source merging § 5.4.1 and producing cross-neighbour tables § 5.5.

- for programmes where synoptic data is taken in many bands with the requirement that several bands are taken together, there is an additional stage which delivers the information in a more useful elegant way, see Figure 7. In this situation, a SynopticSource table is produced by merging detections in different bandpasses taken within a short timeframe using the same approach as a normal source table, see § 5.4.1. The timeframe criterion will be defined for each programme. For WFCAM calibration data this time frame is $\sim$ 15 minutes and typically all broad band filters Z, Y, J, H & K data is taken, although occasionally the narrow band H_2 & Brγ may also be taken. The SynopticMergeLog table is somewhat different from a normal MergeLog since it will contain a timestamp for each frameSet in the SynopticSource table. The SourceXSynopticSource table can be used in the same way as the SourceXDetection table to generate light-curves, but additionally colours at various times can be easily compared. calSourceXSynopticSource would have 50 rows in the example above. This is more than 1000 times smaller than the corresponding setup using calDetectionNeighbours and thus the information is much more easily accessable and much faster to find.

- the master source table can also be used to improve the overall photometry. Assuming that most sources do not vary and that those which vary do so independently of each other, it is possible to recalibrate each individual observation with respect to the master source table. This will only change each zeropoint by a few millimags or centimags at the most, but these differences are crucial to getting the best light curves from this data. The recalibration pipeline discussed in § 5.4.3 can make these changes.

- several variability measurements will be added to the Source table. In each synoptic bandpass, the mean, root mean squared and median absolute deviation are calculated, along with the number of detections, and missing detections. A missing observation occurs when an observation was taken, but no object was detected for this source. This could occur in a number of ways - a moving object, an eclipsed object, a faint object that drops below the threshold in some observations, a nova that was detected when it rapidly brightens, but then fades below the threshold. Using data from the SynopticSource table the Welch-Stetson statistic is calculated. This assumes that the colours of objects do not change and so it is more reliable for some transient variables where bad detections can lead to an incorrect classification. All variability measurements are combined to create a variability classification. For those which are classed as definite variables the main period and number of periods will be calculated.

- a further advantage of this new database model is that the X-neighbour tables can be matched to a greater radius than the simple DetectionNeighbours table without fear of producing unweildly tables with long lookup times. This allows greater flexibility in finding moving sources and opens up more parameter space for searches.
- the Source table can be created early on from a small number of observations ~ 5 per synoptic band. This will contain all objects that can have their variability reliably measured, so it does not have to be changed later, except for the variability statistics which will be updated each time. Therefore the Detection, SynopticSource and SourceXSynopticSource will all increase in length over time, but Source will remain essentially the same unless the extraction or source merging software undergo a radical overhaul.
- non-variable objects can also be selected. The light-curves of existing standard stars are important to test the reliability of the data and new photometric standards can be located using these data.
- although these haven't been shown in Figs 6 or 7, listRemeasurement entities will also be produced for synoptic data.

5.4.3 Science archive photometric/astrometric calibration data

Both astrometric and photometric calibrations data will be associated with images stored in the VSA, and then that same calibration information will be associated with catalogued sources through the latter's association with their respective images. Calibration data present an additional complication in that versioning over time is required, and all previous calibration versions need to be retained (a requirement from the SRAD). Version entities are implied for this purpose, these entities have the following attributes: unique version number, start time and end time.

The initial photometric and astrometric calibration is achieved using 2MASS data. In the WSA this was found to be sufficient to reach the pre-survey goal of 0.02 mag rms photometry and the astrometric calibration is better than 100 mas [10]. Subsequent recalibrations of the photometry may either come from updates from reanalysis of the comparison to 2MASS (as has happened with Y and Z band data in the WSA) or other data sets or by analysis of the photometric standards using a photometric model. Some VSA data may overlap with WSA data, so WSA data may be useful for calibration of VSA data.

The photometric recalibration is shown in Figure 8. The latest zeropoints are stored in the Multiframe and MultiframeDetector tables. The previous zeropoints are then stored in the previousMFZP and PreviousMFDZP tables. The version number with start and end date is stored in PhotCalVers. The udates to the photometric must be passed to the catalogue data.

VDFS-v4 will also include a task to fit a simple model to all the standard star calibration data and also to overlaps between frames. The entity-relation model for this is still being developed.

Figure 9 illustrates multiframe astrometry recalibration entities. The differences between the photometric recalibration and astrometric recalibration are that there is a separate CurrentAstrometry table for the latest calibration. and that the astrometry data is only associated with the detector tables, not the primary metadata as well. The right ascension and declinations in the catalogues will also need to be updated, but these tables have been left out of the figure to save space.

5.4.4 Archive-end flagging of catalogue Data in VISTA

The catalogues will contain various objects which are either detection artifacts (i.e. not real astronomical sources) or are real sources that have detection errors which will cause inaccuracies in measured

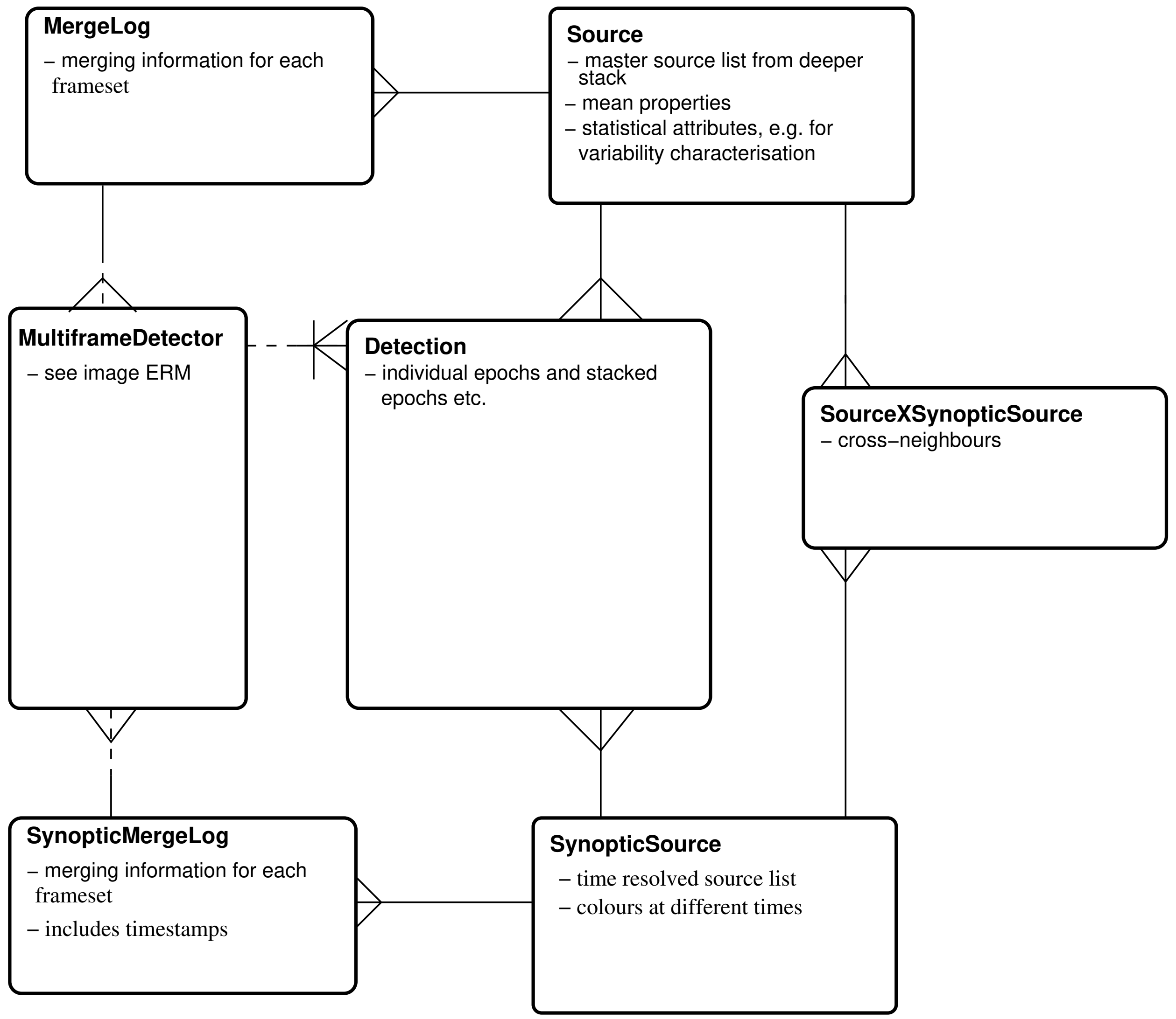

Figure 7: *ERM for multi-band synoptic data in VSA/WSA programmes.*

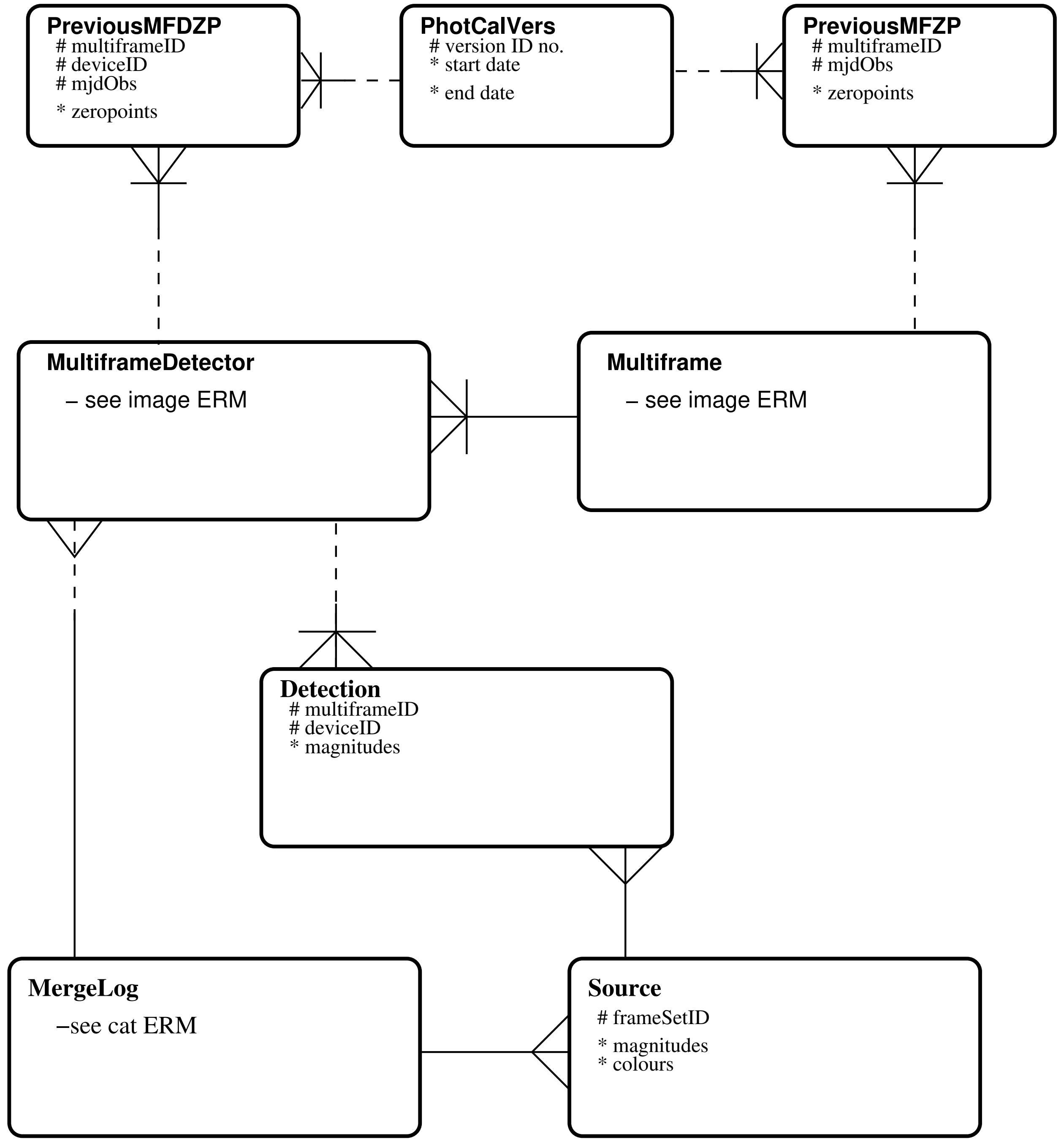

Figure 8: *Entity–relationship model for photometric re-calibration of VISTA multiframes*

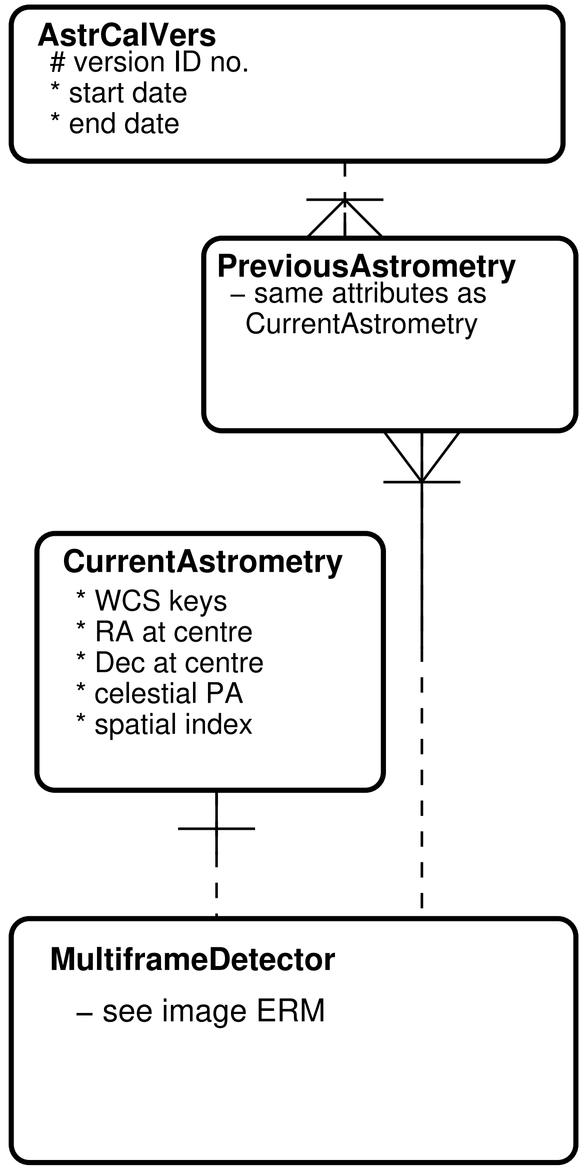

Figure 9: *Entity–relationship model for astrometric re-calibration of VISTA multiframes.*

parameters. These objects need to be identified and flagged. Flagging is much better than deletion for the following reasons:

- mistakes can occur in processing that lead to the flagging or deletion of the wrong objects. It is much easier to correct a flagging error than a deletion error.
- an inaccurate detection may be of some use - the position may be useful for moving objects even if accurate photometry is impossible.
- improved testing can lead to changes in flags. For instance, the definition of near to a bright star may change if it is shown that the extra noise causes no serious problems at a radius where objects had previously been flagged.

The following issues were deemed to be important for flagging in WFCAM data:

- (residual) cross-talk image or affected by same;
- spurious persistence image;
- scattered light halo / diffraction spike / satellite trail;
- incomplete image (pixels hit edge of image area);
- saturated image;
- near edge (e.g. within dither offset) of image area;
- near very bright star;
- one or more apertures of image area;

These issues are ranked in order of perceived relative importance. Most of these issues will also occur in VIRCAM data, although the Raytheon VIRGO detectors used by VIRCAM have some important differences from the Rockwell Hawaii 2 detectors used by WFCAM. These differences may reduce (or possibly eliminate) some of the issues in WFCAM (e.g. cross-talk and persistence) but may introduce other issues instead. These will not be known until the instrument is commissioned.

There are also important differences in the pipeline processing which may lead to more flags in the VISTA data. The VIRCAM data will use a six point dither pattern that produces a $16k \times 13k$ tile. Most pixels in this tile will have been sampled twice, with some 3, 4 or 6 times. However, about 15% will have only been sampled once and therefore will be prone to detector defects, cosmic rays and hot pixels. All the objects in these regions in the shallow surveys will be flagged as near the edge - within one dither offset. A second observation of the same object where two mosaics overlap would make each observation more reliable.

5.4.5 ERMs for the VPSP programmes

The VPSP are described in the Science Requirements Analysis Document . Figure 5 shows the ERM for the both the wide–angle, shallow (unstacked) survey programmes, such as VIKING, VHS and VGPS and also for deep (stacked) surveys such as VUUDS, VIDEO, ELVIS and ULTRA-VISTA. Each survey will have a separate merge log, detection table, source table and remeasurement tables, since each survey has different requirements on filters and number of passes. For instance ELVIS will require the use of a narrow band filter 1.18μm, VUUDS will only require the bluest filters Z,Y&J, and VGPS will require 3 epochs in K_s for proper motion analysis. Most surveys will require galactic extinction corrections, but VGPS will not since almost all objects observed will be well within the galaxy.

5.4.6 The VPSP VMC and VVV programmes

The VMC and VVV surveys both require the use of synoptic data (see Figure 6 for the ERM). In both cases, large areas are required to be observed, and multiple visits (12 in the case of the VMC; 50-200 in the case of the VVV) are also required.

The strategies of the two surveys are quite different. The VMC plans to observe tiles in each of its 3 filters once (Y, J and K_s) as close as possible to minimise variability effects on colours. The VMC will require a synopticSource table with a combination time of ~ 5 minutes.

The VVV team wants to conduct the synoptic survey in J or K_s in its 3rd year, depending on the results of earlier maps that they have produced in the previous two years. These maps will require deep stacks to be produced in the K_s band at WFAU, but the synoptic survey will only be in one band, so a sourceXDetection table is required, but a synopticSource table is *not* required. The VVV Source table will also require 2 epochs in K_s (year 1 and year 2).

5.4.7 Naming of catalogue sources in the VSA & WSA

The current Source tables in the WSA and future Source tables in the VSA may be updated in the future to contain an *IAU Object Name*, an IAU approved name. For UKIDSS objects the standard names will be:

UXXX JHHMMSS.ss+-DDMMSS.s

where XXX is the survey name: LAS, GPS, GCS, DXS, UDS

For instance an object at 12h56m30.04s -01d13m46.7s in the LAS would have an *IAU Object Name* ULAS J125630.04-011346.7

The acronyms for VISTA PSPs will be decided in due course, once the final surveys are selected. These will have similar names to UKIDSS objects.

5.5 Non–VIRCAM data (catalogues and images)

Figure 10 illustrates the general relational model for non–VISTA data that will be held in the VSA for the purposes of joint querying. Examples of non–VISTA data that are required to be held (see the SRAD; AD01) are legacy photographic all–sky surveys (e.g. SSS, USNO); current optical and infrared surveys (e.g. SDSS, 2MASS, MGC); complementary optical imaging surveys for the VISTA PSP programmes; and finally survey catalogues from more distant and diverse wavelengths (e.g. FIRST; ROSAT–ASS). The entities in Figure 10 are described and related as follows:

- for each non–VISTA survey, a catalogue will be stored (e.g. SkyServer.PhotoObj from the SDSS or the SSA merged source catalogue);
- a table of 'cross–neighbours' will be created for every source in the non–VISTA list that is near to a given VISTA source in the merged source detection list for the appropriate programme(s). The maximum angular radius for neighbours will be set according to appropriate science drivers, e.g. expected proper motion displacement between the catalogues, or in the case of large error boxes on the non–VISTA data, a 5σ positional tolerance. Every VISTA source may have one or more non–VISTA cross–neighbours; likewise every non–VISTA source may be cross–identified with one or more VISTA sources; the UID of the cross–neighbour pair is the combination of UIDs in the VISTA and non–VISTA source lists;

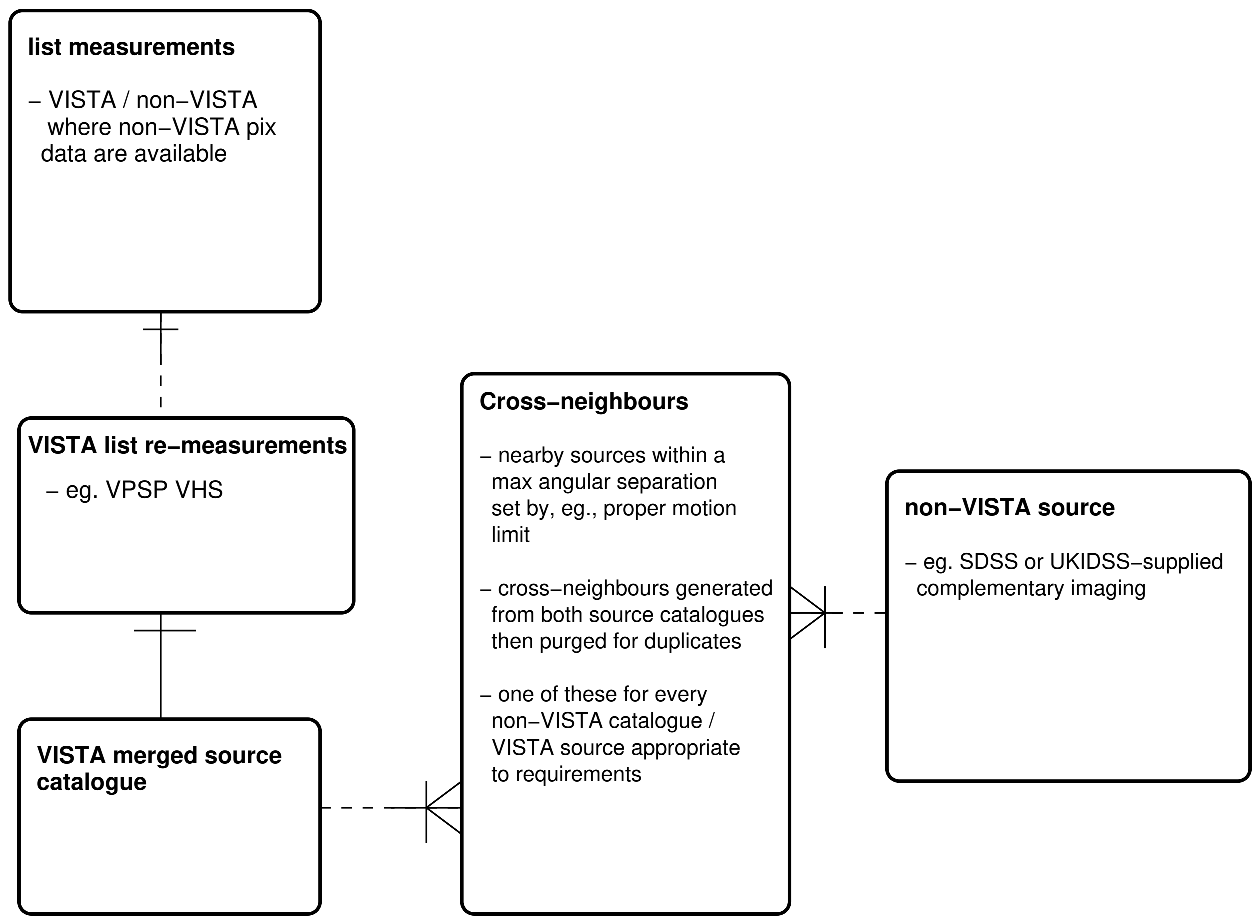

Figure 10: *Generalised entity–relationship model for non–VISTA survey data that will be held in the VSA.*

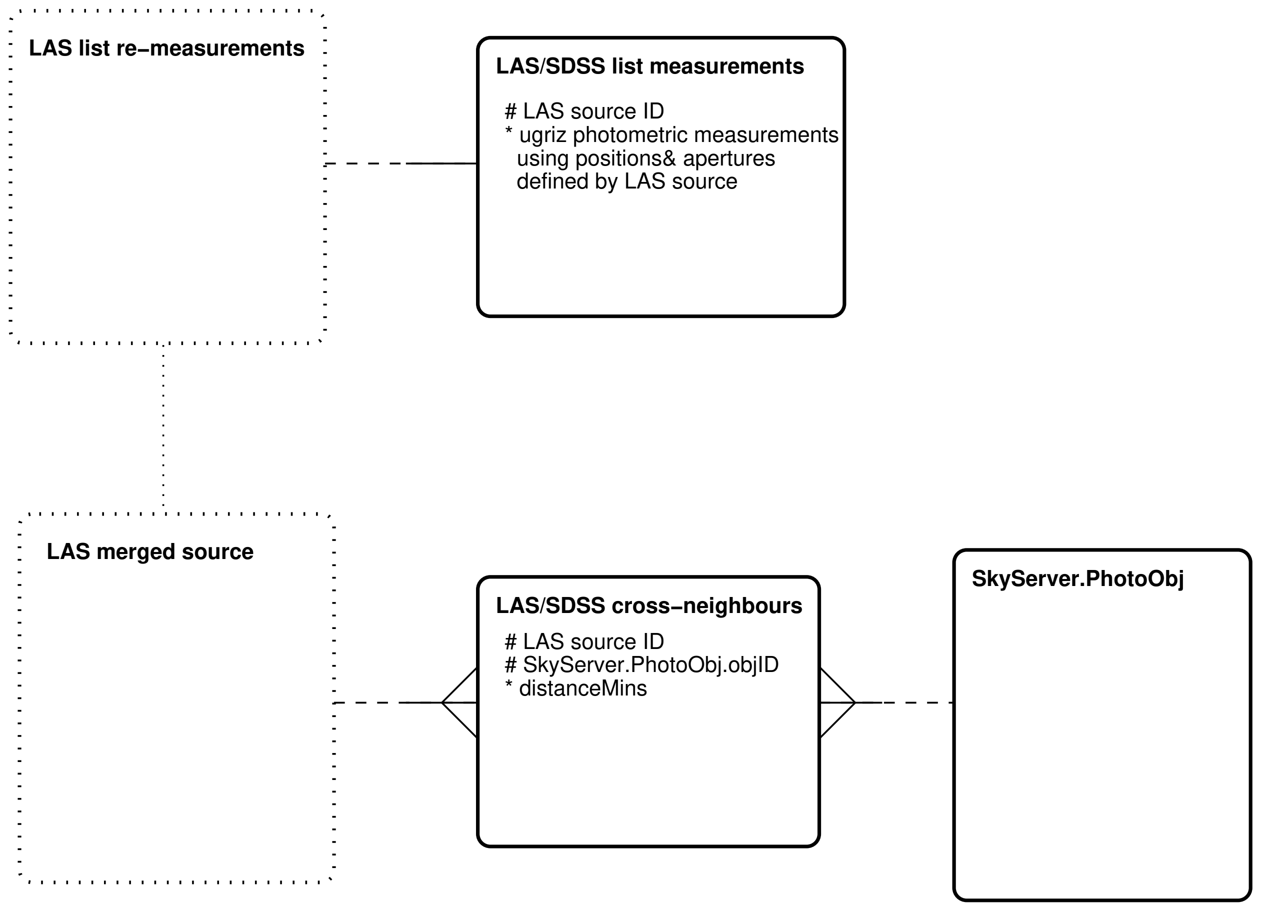

Figure 11: *Entity–relationship model for the UKIDSS–LAS/SDSS surveys.*

5.5.1 Example: the UKIDSS LAS and its relationship to the SDSS

The relationship between the UKIDSS LAS and the SDSS is illustrated in Figure 11 as an example of the connection between a programme of WFCAM data and a non–WFCAM external survey dataset. This ERM applies equally to VISTA surveys and external data.

5.6 Curation ERMs

Figure 12 details the curation entities that will be kept (in addition to the merge logs associated with each programme, discussed previously). These form a 'star schema' around the central archive entity that describes each archived programme. The entities are as follows:

- a set of Curation tasks (described in the next Section) is defined for the archive;
- each run of a curation task is recorded in the ArchiveCurationHistory entity along with a log file and timestamp. The cuEventId is the record of each task.
- the curation history of each programme is recorded in ProgrammeCurationHistory
- each programme is related to a survey which is released. In the WSA for instance the 5 UKIDSS programmes are grouped together to form the UKIDSS release, which is proprietary. These 5 programmes are also grouped together with the Planet Transit Survey and the calibration data into a WORLD release which is non-proprietary, but is released 18 months later than the

UKIDSS release. A similar set up may occur with the VPSP which will be released first to astronomers in ESO countries and later to the rest of the world. Information about each survey is in Survey, the relationship between each survey and the programmes is in SurveyProgrammes and the release history of each survey is in Release.

The other entities in Figure 12 help define the actions of some of the curationTasks:

- RequiredNeighbours defines links between source and detection tables in VISTA programmes and similar tables in external surveys. This is the basis of CU16;
- the required combination of filters for each programme is defined in an entity (RequiredFilters which references filter descriptions.
- RequiredStack and RequiredMosaic in Figure 4 are used to define the action of CU13 and CU14. These entities define what final deep products are required, specifying the filter, position, stacking and extraction information.

More details of the curation procedures and software are given in Section 8.

5.7 Arrangement of entities within the archive

The previous Sections describe the various entities that will be used to store and track information in the VSA. However, in Section 5.1 we described the requirement for several independent databases to archive open–time observations, to ensure proprietorial rights for all data are protected, and yet to allow unfettered 'world' access to the international astronomical community once proprietorial periods have expired. The various databases needed to meet these requirements will be arranged as follows:

5.7.1 Internal, incremental database

WFAU will curate an internal access–only database using a 'load server' (see HDD; AD05). Copies of all entities discussed previously will be held in this database, which will ingest new data on a daily basis.

5.7.2 Open time observations databases

Open time programmes will be curated in the same way as the large–scale public surveys within the offline database. Merged source catalogues will be produced where possible and where required by the PIs. Once all the data for a particular programme are ingested and merged into the relevant programme entities within the archive, they will be copied into an individual database on the public server with access restricted to the PI via password protection.

5.7.3 Public survey release database

All internally curated VPSP entities will be periodically frozen, backed up and released by copying into a VPSP release database on the public–access catalogue server.

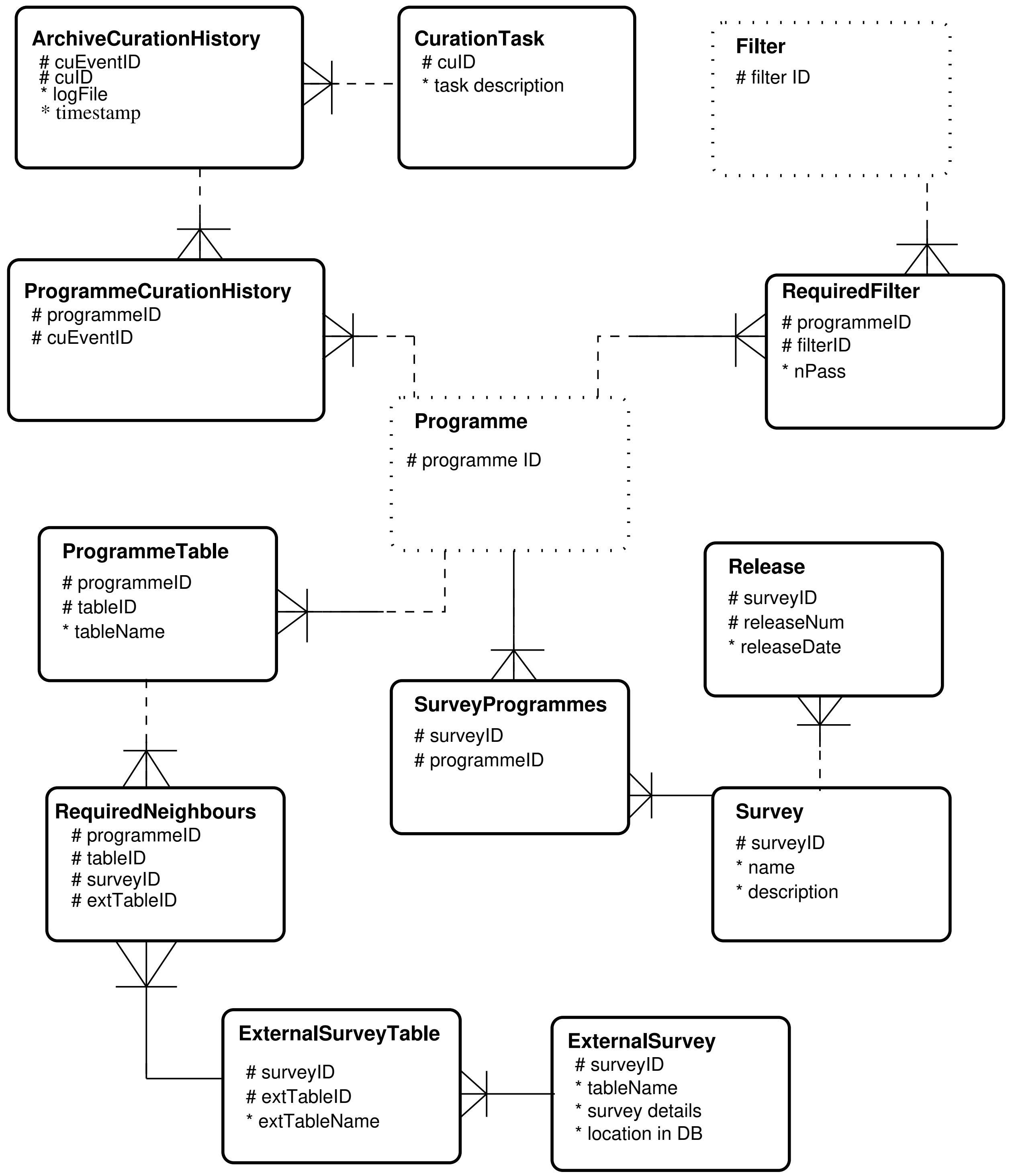

Figure 12: *Entity–relationship model for curation information within the WSA.*

5.7.4 World readable database

All data will ultimately become world–readable as proprietorial periods (where they exist) expire. When a significant amount of data have become world readable, a 'world' release will be made into a database with unrestricted access. This will involve creating world readable subsets of all currently released VPSP database catalogues in the world database based on the present release date, the observations date and the proprietorial period and copying in open–time catalogues that have passed their proprietorial expiration. Table 'views' will be created (see the next Section) to create a logical overview catalogue that can be queried for general interest usages not aimed specifically at a particular programme.

6 Tables, fields, keys and views

The VSA will be implemented in SQL Server. This Section describes the details of the database schema files via examples, and illustrates the WSA databases. The following conventions have been adhered to in generating schema files:

- SQL scripts are used to create the database tables, and these scripts are placed under CVS version control;
- meaningful table names that relate directly to the ERM descriptions in the previous Section are employed, but excessively long table names are avoided;
- attribute names are kept meaningful, and correspond as closely as possible to FITS key names in the progenitor FITS files where possible and appropriate;
- attribute units are 'natural' so as to be clear to users – e.g. arbitrary scaling to fit attributes into smaller datatypes (for example storing magnitudes in millimag units in 2–byte integers instead of normal magnitude units in 4–byte reals) is not be done;
- where possible, meaningful unique identifiers are used to track the provenance of a record – e.g. VISTA images will have compound UIDs made up of a CASU or WFAU prefix (CASU for observed and pipeline processed images; WFAU for deep stacks, mosaics and differences images created by WFAU) from original VISTA run numbers (or compounds of productIDs and filterIDs in the case of WFAU products), date observed (or date created in terms of WFAU products).

6.1 SQL Server schema SQL scripts

An example extract from the SQL Server VSA schema SQL scripts is as follows:

```
CREATE TABLE Programme(
-------------------------------------------------------------------------------
--/H Contains details of the VISTA observation programmes.
--
--/T This table contains details of the observation programmes undertaken
--/T with VISTA and for which observations are stored and curated in the VSA.
--/T Names of the tables that store information for each programme are
--/T recorded here.
--
--/T The unique programme ID number (assigned internally within the VSA)
--/T identifying all VISTA observation programmes is as follows:
```

```
--/T <b>programme ID: description</b>
--/T         0: not a programme: used for calibration/confidence frames common to many programmes
--/T         1:  Commissioning data
--/T       101:  VPSP Magellenic System Survey (VMC)
--/T       102:  VPSP UKIRT Ultra Deep Survey (VUUDS)
--/T       103:  VPSP Variables in VL Survey (VVV)
--/T       104:  VPSP Kilo-Degree Galaxy Survey (VIKING)
--/T       105:  VPSP Deep Extragalactic Survey (VIDEO)
--/T       106:  VPSP Hemisphere Survey (VHS)
--/T       107:  VPSP Emission Line Galaxies Survey (ELVIS)
--/T       108:  VPSP Galactic Plane Survey (VGPS)
--/T       109:  VPSP Observing Beyond Reionisation (ULTRA-VISTA)
--/T    1000+: Open Time programmes (PI)
--/T   10000+:  Service programmes
--
--/T Required constraints: primary key is (programmeID)
--------------------------------------------------------------------------------
programmeID       int not null,          --/D UID of the archived programme coded as above  --/C ID_SURVEY
title             varchar(32) not null,  --/D a short title for the programme, e.g.\ "VPSP Magellenic System Survey"   --/C NOTE
description       varchar(256) not null, --/D a concise description of the programme   --/C NOTE
reference         varchar(256) not null, --/D a reference for the programme, e.g.\ "http://www.eso.org/"   --/C REFER_CODE
propPeriod        int not null,          --/D the proprietory period for any data taken for this programme in months, e.g.\ 12 for open time. --/U months  --/C TIME_PERIOD
dfsIDString       varchar(64) not null,  --/D The description used within the data flow system (i.e. the value of the appropriate FITS keyword) --/C ??
catalogueSchema   varchar(64) not null,  --/D Script file that describes the catalogue schema --/C ??
detectionTable    varchar(64) not null,  --/D Table name of detections --/C ??
sourceTable       varchar(64) not null,  --/D Table name of merged sources --/C ??
listRemeasTable   varchar(64) not null,  --/D Table name of individual passband list re-measurements --/C ??
sourceRemeasTable varchar(64) not null,  --/D Table name of co-located list re-measurements --/C ??
mergeLogTable     varchar(64) not null,  --/D Table name of curation log for source merging --/C ??
frameTable        varchar(64) not null,  --/D Name of table containing frame metadata --/C ??
detectorTable     varchar(64) not null,  --/D Name of table containing frame detector metadata --/C ??
frameSetTolerance  real not null,        --/D The match tolerance for different passband frames --/U Degrees --/C ??
pairingCriterion  real not null,         --/D The pairing criterion for associating detections into merged sources --/U Degrees --/C ??
detectionObjID    varchar(64) not null,  --/D The attribute name(s) of the UID for detections  --/C ??
sourceObjID       varchar(64) not null,  --/D The attribute name(s) of the UID for merged sources --/C ??
rmOldFrameSetsSQL  varchar(64) not null,  --/D SQL script name for deprecating old frame sets during source merging --/C ??
adjacentFrameTolerance  real not null,    --/D The match tolerance for adjacent frames in the survey --/U Degrees --/C ??
neighboursSchema  varchar(64) not null,  --/D Script file that describes the neighbour tables schema for this programme --/C ??
CONSTRAINT pk_Programme PRIMARY KEY (programmeID)
)
GO

CREATE TABLE Filter(
--------------------------------------------------------------------------------
--/H Contains details of the VISTA filters.
--
--/T The unique ID number (assigned internally within the VSA)
--/T identifying all the filters available in VISTA.
--/T
--/T Required constraints: primary key is (filterID)
--------------------------------------------------------------------------------
```

```
filterID     tinyint not null,       --/D UID of combined filter (assigned in VSA: 1=Z,2=Y,3=J,4=H,5=Ks,6=J118,8=blank)    --/C INST_FILTER_CODE
shortName    varchar(4) not null,    --/D Short identification name for the filter --/C ??
name         varchar(16) not null,   --/D The name of the filter, e.g.\ "MKO J", "VISTA Y" etc. --/C NOTE
description  varchar(256) not null,  --/D a concise description of the filter   --/C NOTE
cutOn        real not null,          --/D The 50% cut-on wavelength of the filter --/C ?? --/U microns
cutOff       real not null,          --/D The 50% cut-off wavelength of the filter --/C ?? --/U microns
aebv         real not null DEFAULT -0.9999995e9,  --/D The constant to multiply the Schlegel map E(B-V) by to get the total galactic extinction.  --/N -0.9999995e9
vegaToAB     real not null DEFAULT -0.9999995e9,  --/D The constant to convert Vega magnitudes to AB magnitudes.  --/U mag  --/N -0.9999995e9
CONSTRAINT pk_Filt PRIMARY KEY (filterID)
)
GO
```

All tables and attributes will be defined in this way. Attributes have datatypes as specified; a short comment in the script gives additional description, comments and/or units as appropriate. Appendix 9.3 gives more detailed examples for the VSA multiframe, and VPSP VHS tables which may reside in the database.

6.2 Constraints

Constraints (in the form of primary keys and foreign keys) will again be specified via SQL scripts. An example is given in Appendix 9.4 which details the constraints on the tables created using the scripts in Appendix 9.3. The resulting 'star schema' picture of the tables and constraints as implemented in SQL Server, and ready for data ingest, can be seen on the science archive website [12].

6.3 Views

Views are *logical* tables (rather than physical tables on disk) expressed via an SQL script that is executed when a query is requested on the view. WSA views can be located in the Schema Browser on science archive [12]. Some example views in the VSA are as follows:

- A view will be created of a seamless reliable selection of the sources consistently identified as extended sources in the VPSP VHS;
- A view will be created of a seamless reliable selection of the sources consistently identified as point sources in the VPSP VHS;
- A view will be created of all point sources in the VPSP with 5 or more observations for variability studies;

7 Indexing and other implementation details

7.1 Spatial indexing attributes

One of the main types of query that any astronomical database will encounter is one making use of spherical co-ordinates to locate a small sky region of interest – for example, several usages in the SRAD require position/proximity searches. A primary requirement on database organisation is therefore sorting or some other form of spatial indexing. Many potential schemes are possible, but the VSA will employ the Hierarchical Triangular Mesh (HTM) indexing scheme [1] employed in the SDSS and other current surveys.

Briefly, HTM employs a hierarchical quad–tree indexing scheme to code up the positions of nested equilateral triangular patches of sky of decreasing size, starting with 8 triangles covering the whole sky, 4 octants in each hemisphere. The VSA will employ an 8–byte integer index for up to 20 decimal digits which will uniquely specify a position to a resolution of ~ 10 milliarcsec. Where any celestial co-ordinate attribute pair occurs in a table, a corresponding HTM index will also be present (see, for example, table `Source` in Appendix 9.3).

7.2 Table indexing

To greatly enhance query performance on commonly queried attributes, (e.g. position, magnitude, colour etc.) table indices will be created. These indices are employed by the query optimiser at query run time to avoid the DBMS having to search entire tables to execute the query – this is of course standard practice in RDBMS design.

Table indexing is not an exact science, and a certain amount of experimentation will have to be undertaken with real–world VSA data and queries to ellucidate and optimise the nature, and number of the indices created for each table. We will use the experience gained from the WSA and the following general rules that will be adhered to in creating indices.

SQL Server uses B–tree (B≡balanced) indexing, and there are two basic types of index: clustered and non–clustered. A clustered index results in the table data being reordered (i.e. sorted) on disk, so only one clustered index can be made per table; otherwise any number of non–clustered indices can be made (subject to disk space limitations). Both types of index can be constructed on multiple attribute sets. In creating the VSA indices, we will adhere to the following general principles:

- when building an index on more than one attribute, the attributes will be put in order of decreasing selectivity;
- the default–generated clustered index will be taken as that based on each table's primary key – SDSS experience [3] and WSA experience is that this is the optimal set–up (for example, the primary key of detections is an UID based on a progenitor frame ID, and hence detections on one frame, which will tend to be accessed together during both curation and use, will be stored together on disk);
- for performance, indices should ideally be held on a different physical disk volume to the data themselves so that the index read is as close to 100% sequential as possible;

The HTM attribute will be included as one of the most selective of each table's attribute set in at least one index, where appropriate. We have gained much experience with indexing the WSA, which is built upon very similar lines as the VSA.

8 Details of curation procedures and software

For each of the curation tasks identified in the use cases presented in Appendix 9.1, we have followed standard practice in the rational unified process and identified the primary scenarios associated with each case. These are presented on a case–by–case basis in the following sections in the form of a numbered–steps pseudo–code description along with an activity diagram illustrating the flow of events. Figure 13 shows a key for the diagrams, while Figure 14 illustrates the overall logical flow of the curation tasks (which, within any given curation period, run sequentially). For some of the more complicated cases, algorithmic details are given. Implementation details are given for all tasks.

Some general notes on the curation tasks are as follows:

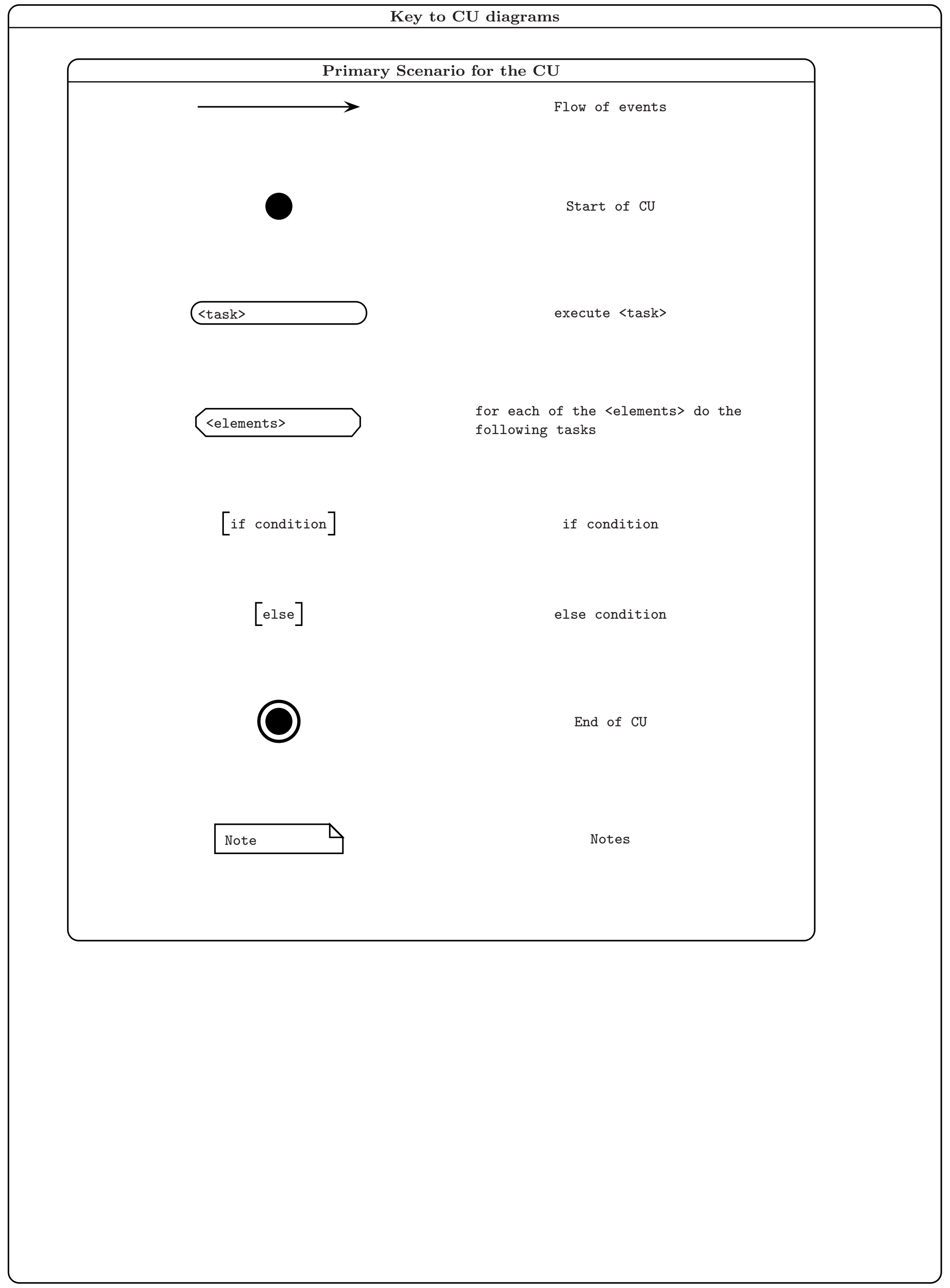

Figure 13: *Key to the curation activity diagrams.*

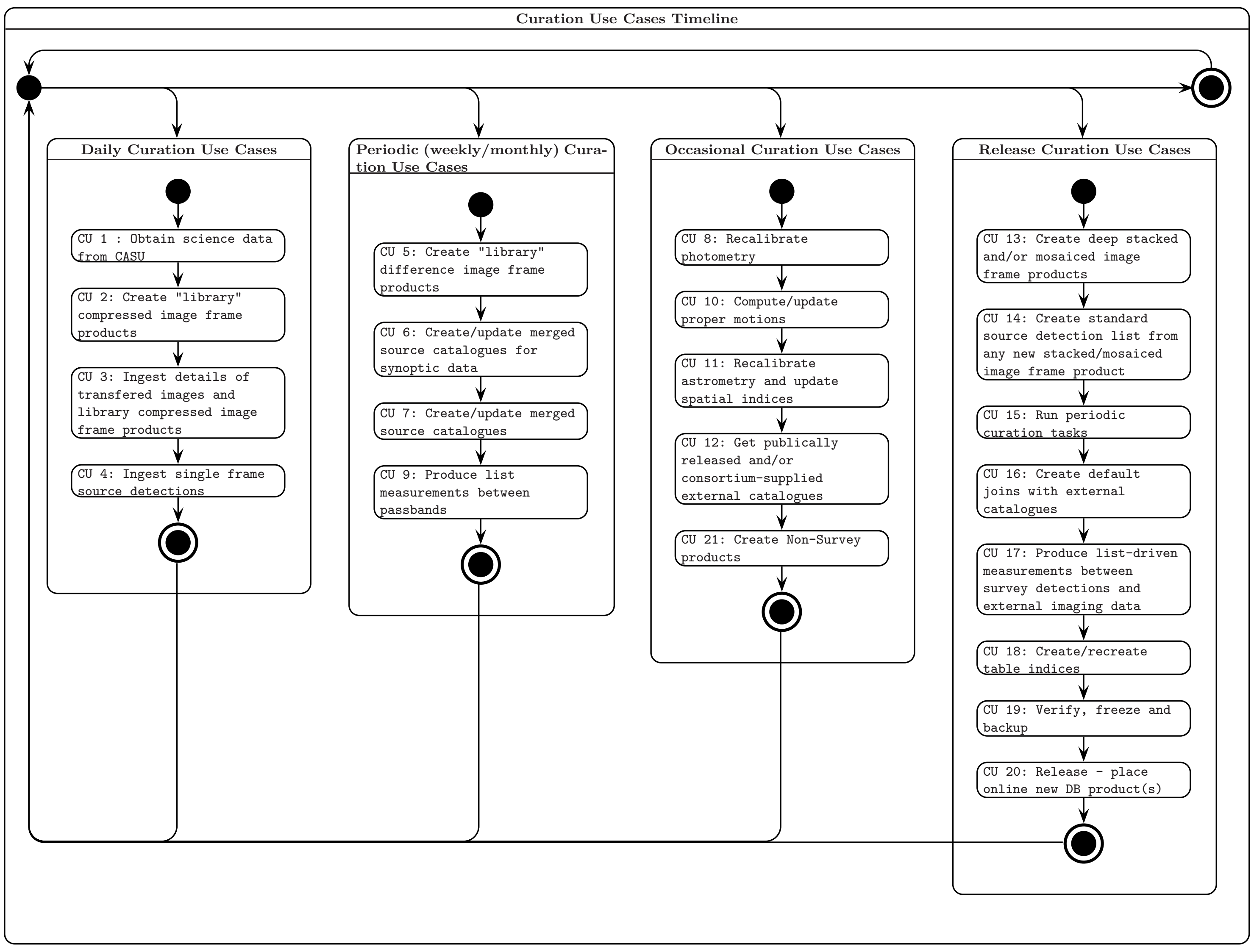

Figure 14: *Top–level activity diagram for the curation tasks.*

- all curation tasks are implemented using Python scripts to 'glue' C/C++ applications (existing, supplied or in some cases specially written);
- most tasks execute on the Linux server side (those involving to data transfer/ingest and pixel manipulation) while a few execute on the DBMS load–server side. For details of the hardware configuration, see the Hardware/OS/DBMS design document (AD05);
- several of the curation tasks (e.g. CU3) involve the transfer of bulk data (anywhere from Mbytes to a few Gbytes in size) from the Linux server that receives the data from CASU and the Windows/SQL Server 'load' server. This requires communication between processes running on the two machines, which is achieved via NFS file shares.

8.1 CU1 : Obtain science data from CASU

```
CU1 : Obtain science data from CASU : Primary Scenarios

Precondition(s): Any previous transfer is complete;
                 Time is after normal working hours;
                 Network connection closed

Flow of events:

1. Initiate a multithreaded network connection of 8 scp scripts.

2. For each new source directory

   a) Test source directory for read-readiness

   b) For each file in a read-ready source directory

      i) Check VSA filesystem for existing file (rerun duplicate)

      ii) Transfer the file to the destination directory

      iii) Verify that the transfer was successful using an md5 checksum

      iv) log the file transfer in transfer log file

   end

end

3. Close network connection

4. Copy transfer log to permanent area to save a record of all transactions

5. Update DB system curation log for CU1 for all relevant programmes

Postcondition(s): Network connection closed;
                  Intermediate log file closed
```

Implementation details: files will be copied straight to their final destination to avoid further IO over–heads; implementation will be via a Python script running on the mass–storage Linux server side; the CU1 script will be invoked by a Linux shell script ‘cron’ job to automatically execute each night. The final step will communicate with the DBMS server via the Python DBI (database interface) module.

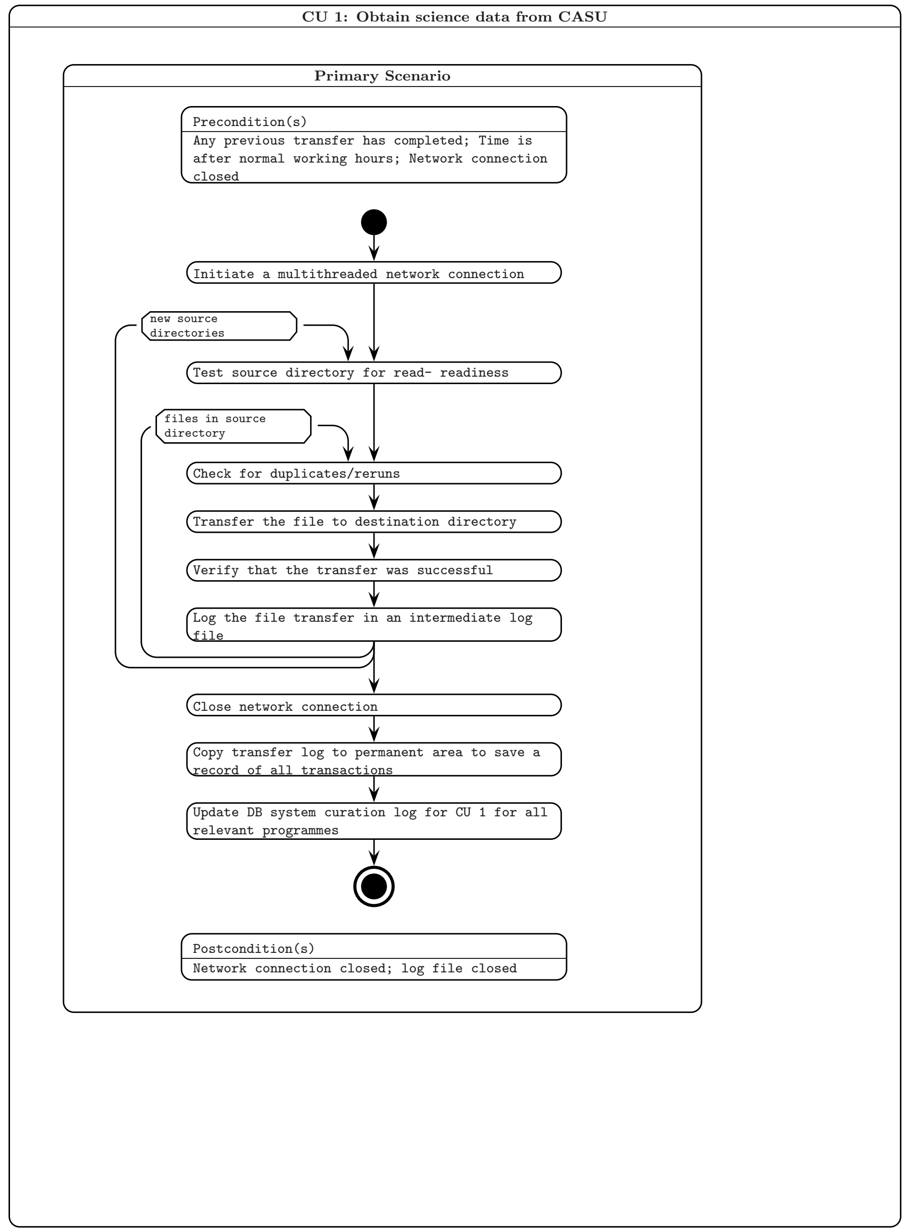

Figure 15: *Activity diagram for curation task CU1.*

8.2 CU2: Create library compressed image frame products

```
CU2: Create "library" compressed image frame products : Primary Scenarios

Precondition(s): Log of recently transfered files exists and is complete;
                 No transfer is currently underway

Flow of events:

1. For each image file listed in the log

   a) For each image array in the image file

      i) Read in image array

      ii) Compress image

      iii) Add axes

      iv) Write compressed file

      v) log newly created compressed image file

   end

end

2. Update DB system curation log for CU2

Postcondition(s): Log file closed
```

Implementation details: this task will again run on the mass–storage Linux server side. Python scripting will be used to 'glue' the necessary procedures together; existing code (e.g. ImageMagick) will be employed to compress FITS to JPEG. Communication with the DBMS for step 2 will be as for CU1.

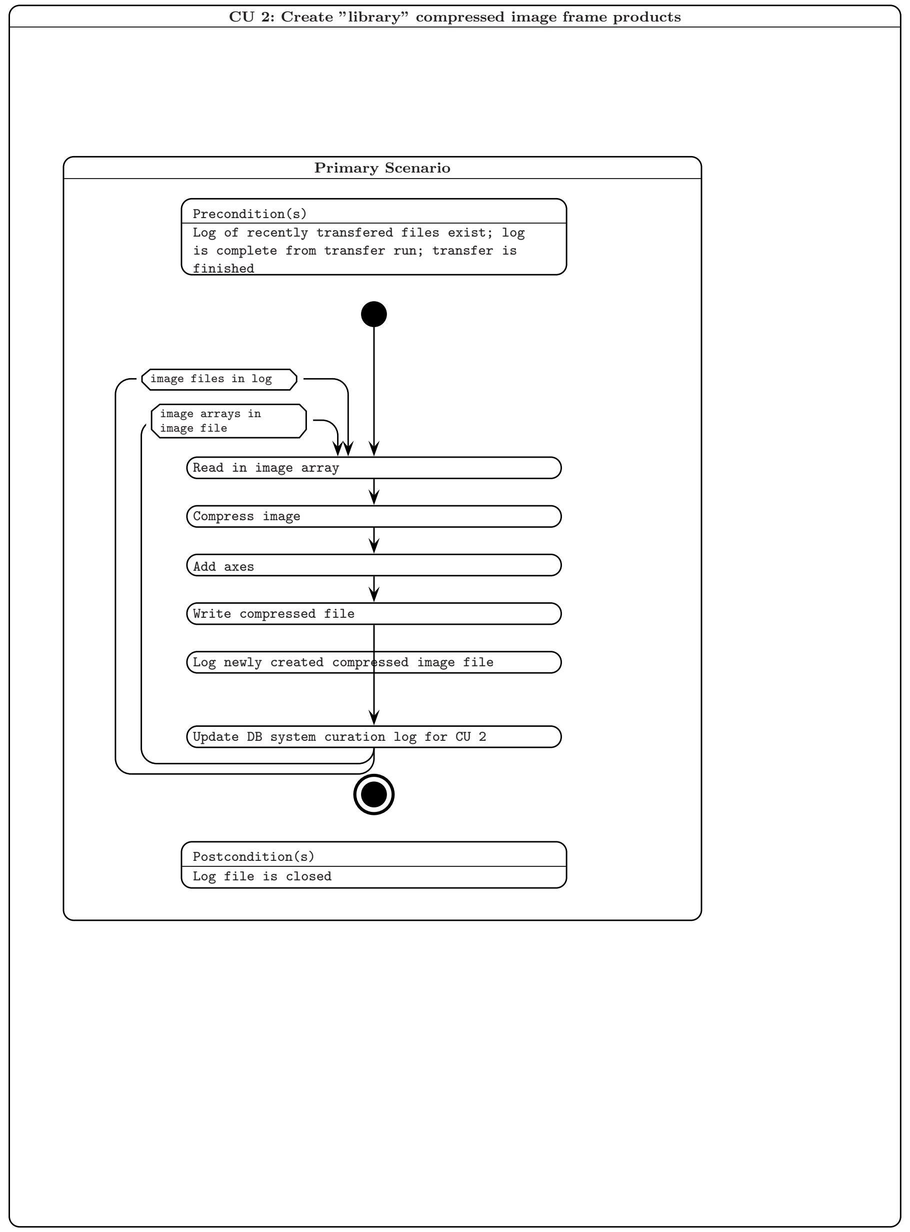

Figure 16: *Activity diagram for curation task CU2.*

8.3 CU3: Ingest details of image products

```
CU3: Ingest details of transfered images and library compressed image frame
     products: Primary Scenarios

Precondition(s): Logs of image products are ready;
                 DB system is ready for updates

Flow of events:

1. Find files that are ready for metadata ingestion in the transfer log.

2. Produce a list of tables that need to be updated with the metadata

3. For each table:

   a) extract the sql information to fill the tables with the metadata.

4. For each set of pixel files and catalogue file in the science frames:

   a) Check the metadata for correct range, not a number values
   b) For each table:
     i) extract the required metadata and write to a csv file

4. Bulk load the csv files into the database.

5. Delete intermediate files

6. Update DB system curation log for CU3

Postcondition(s): DB system is write-unlocked
```

Implementation details: a Python script running on the Linux server side will execute a C/C++ application to strip metadata to intermediate ASCII CSV files. The procedure for bulk loading into the DBMS is a single SQL bulk insert invocation on the server side.

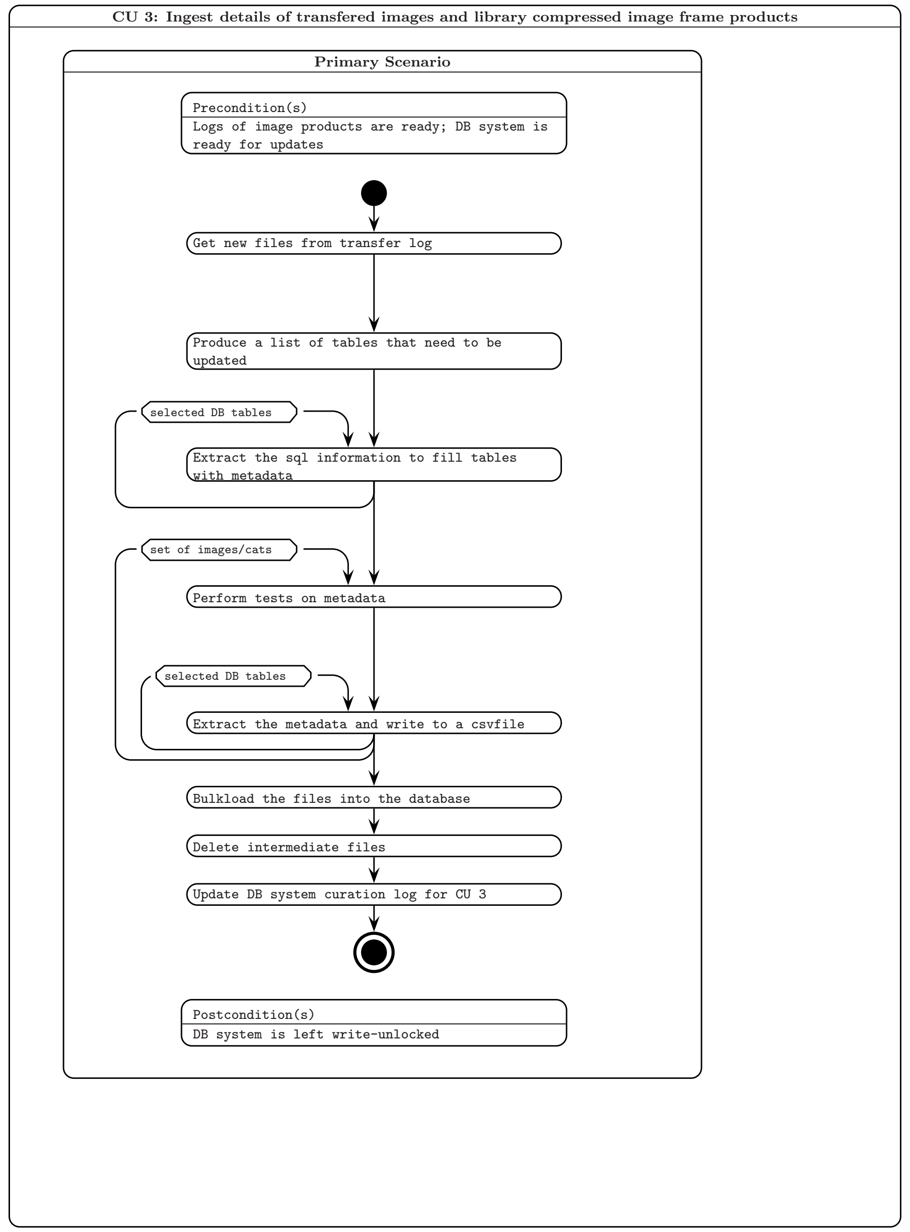

Figure 17: *Activity diagram for curation task CU3.*

8.4 CU4: Ingest single frame source detections

```
CU4: Ingest single frame source detections : Primary Scenarios

Precondition(s): Log of transfered files is ready and complete;
                 DB system is ready for updates

Flow of events:


1. Find files that are ready for catalogue ingestion in the transfer log.

2. For each set of images/catalogs:
   a) Check that the metadata has already been ingested (CU3)

3. Produce a list of tables that need to be updated with catalogue data

4. For each table:
   a) extract the sql information about how to fill the table with the catalogue data.

5. For each catalogue file:
   a) Check the data for correct range, not a number values
   b) For each table:
      i) extract the required data and write to a native binary file

6. For each table:
   a) Bulk load the native binary file into the database.

7. Delete intermediate files

8. Update DB system curation log for CU3


Postcondition(s): DB system is left write-unlocked
```

Implementation details: a Python script running on the Linux server side will execute a C/C++ application to strip FITS table data to intermediate native binary files, which are employed as highly compact, intermediate ingest files for bulk load. The procedure for bulk loading into the DBMS is a single SQL bulk insert invocation on the server side.

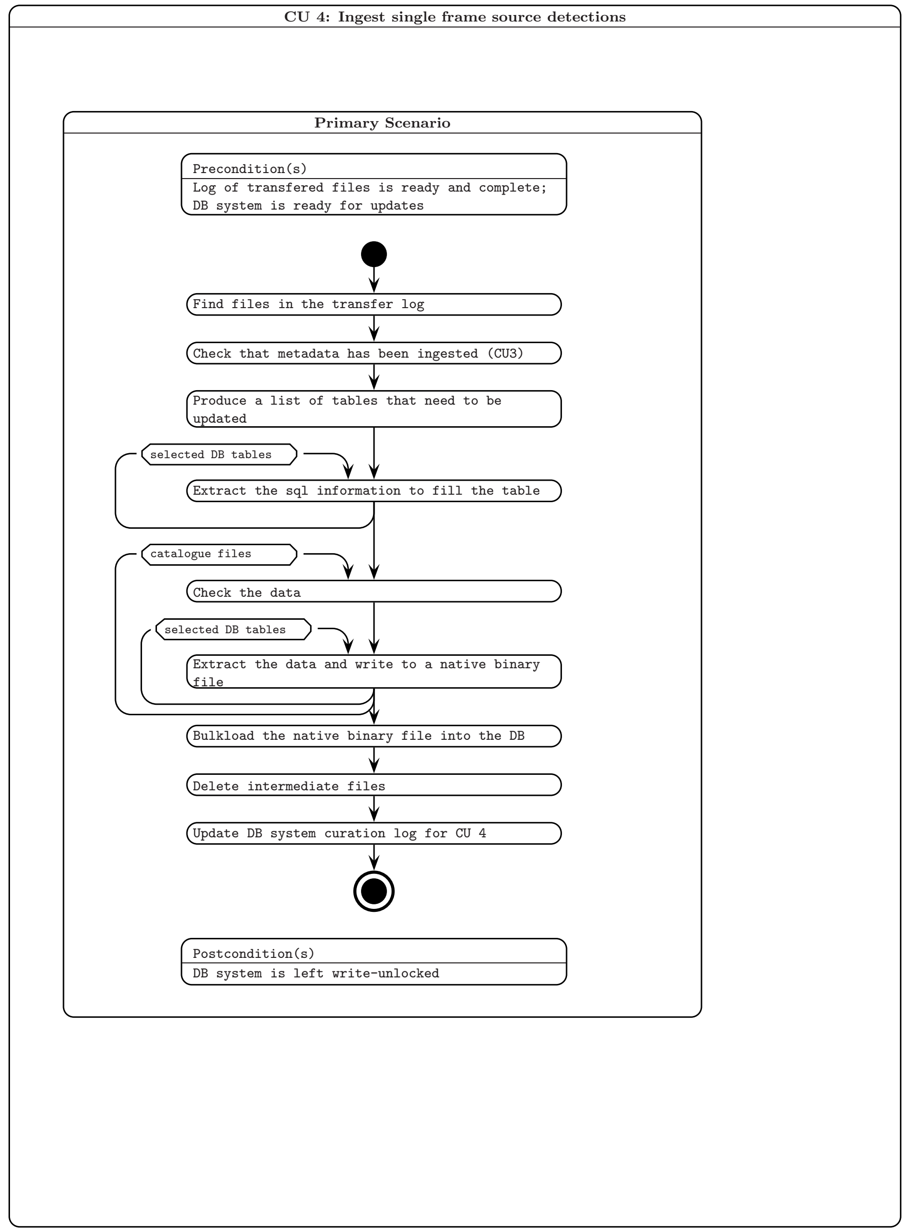

Figure 18: *Activity diagram for curation task CU4.*

8.5 CU5: Create library difference image frame products

```
CU5: Create "library" difference image frame products : Primary Scenarios

Precondition(s): DB system is ready for updates

Flow of events:

1. Find required difference image criteria

2. Query image DB to determine list of new images available which
   match above criteria

3. Find image pairs by position

4. For each image pair:

   a) create difference image and difference confidence image

5. Update DB system curation log.

Postcondition(s): Image DB system is left write-unlocked
```

Implementation details: a Python script running on the Linux server side will use Python DBI to query the DBMS server for images appropriate for default difference imaging (i.e. those J_{118} and J band images from the VPSP ELVIS). The difference image is created using a C code produced by CASU (see [20]). Information concerning the newly created image products will be communicated back to the DBMS – i.e. updates to the appropriate tables in the image database – via Python DBI.

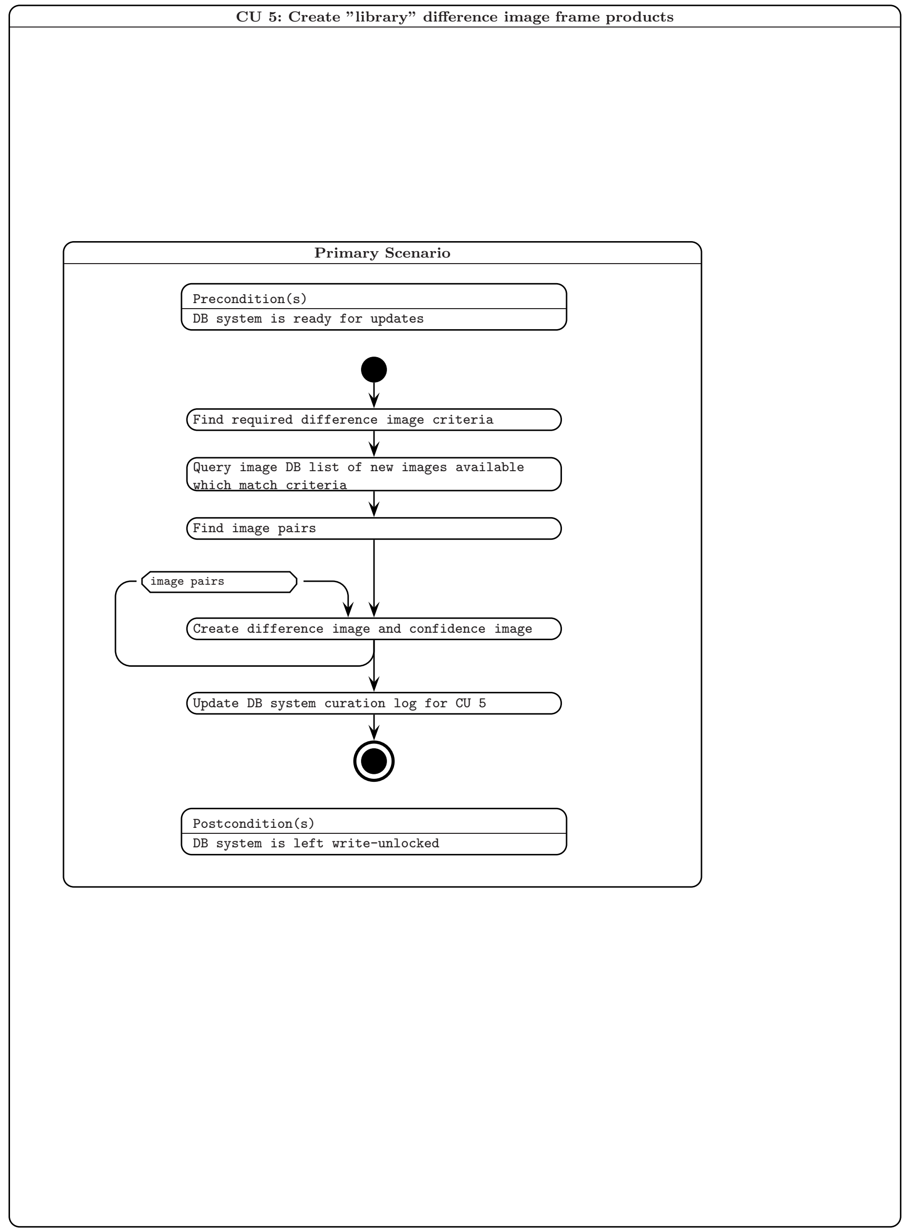

Figure 19: *Activity diagram for curation task CU5.*

8.6 CU6: Create/update Synoptic Source Tables

```
CU6:  : Primary Scenarios

Precondition(s): DB system is ready for update

Flow of events:

1. Find or create (CU13/14/7) a master Source table from deep stack of
several observations.

2. For each observation:

   a) Recalibrate single observation (CU8) w.r.t. master catalogue.

3. Merge multiple passbands (observed over small time range) into
SynopticSource table  (if necessary)

4. Create/update cross-neighbour table between the Source table and
the SynopticSource table or the Detection table (CU16)


5. For each source in master catalogue:
   a) Run variability statistics package to calculate likelihood of
    variability, magnitude of variabiliy and period of variability.

6. Run list driven photometry (CU9) on each observation

Postcondition(s): DB system is left write-unlocked
```

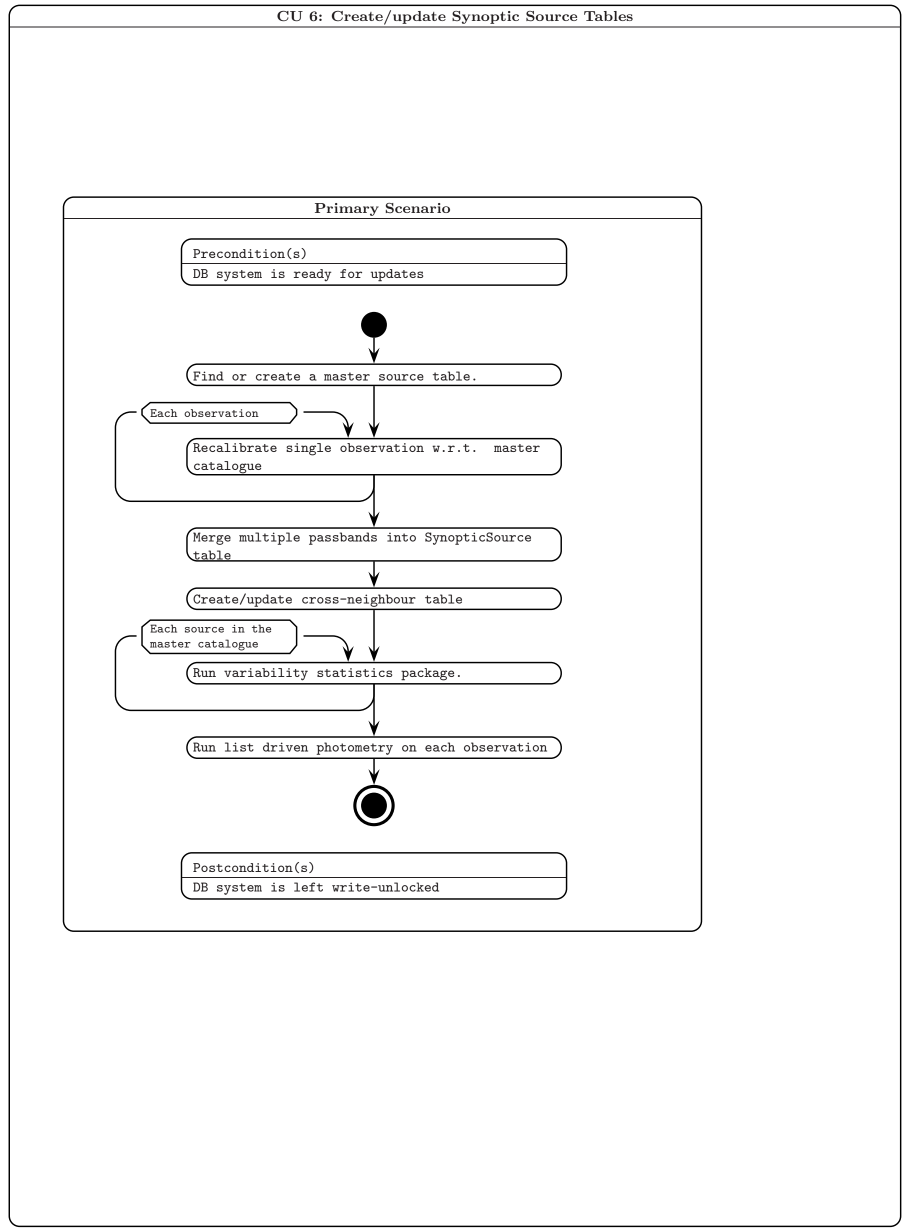

Figure 20: *Activity diagram for curation task CU6.*

8.7 CU7: Create/update merged source catalogues

```
CU7: Create/update merged source catalogues : Primary Scenarios

Precondition(s): DB system ready for update
                 Prescription for merged source catalogue is available
                 (e.g.\ ZYJHKs(x3), no galactic extinction correction
                 for VPSP VGPS)

Flow of events:

1. Find required filters, bandpasses and galactic correction from programme information.

2. Create list of new framesets including existing but incomplete framesets

3. For every frameset in the frameset list

   a) Cycle through the passbands to create a file of detections

   b) Create a merge table file from detection pairs

   c) Copy the data from detections into the merge table file according to SQL
   specifications

4. Remove deprecated sources and mergelogs

5. Ingest merge tables

6. Reseam source table

Postcondition(s): DB system is left write-unlocked
```

Implementation details: the merging algorithm for a set of VISTA detections in different passbands for the same field will follow methods developed for the SSS and used in the WSA: the individual passband lists are sorted; pair pointers are produced between every combination of the passbands taken two at a time where the nearest object out to a maximum pairing radius of 3 arcsec is pointed to. An association algorithm then cycles through each image in each passband in turn, checking for consistent pointers forwards and backwards and keeping track of which detections have been written into the merged list so that each image is merged in only once. This general utility will be implemented such that it can be run for any programme for which a passband prescription set is specified in the RequiredFilter table (see Figure 12).

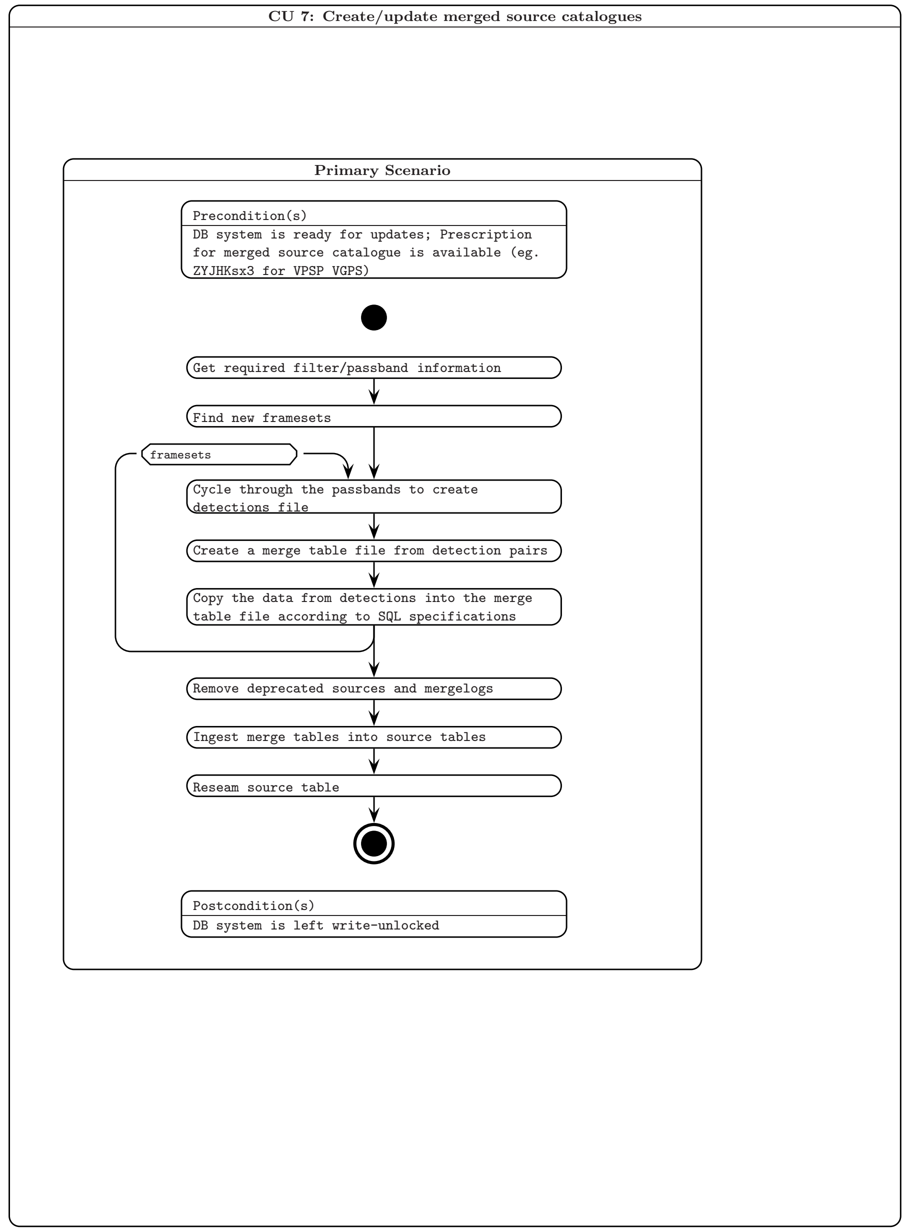

Figure 21: *Activity diagram for curation task CU7.*

8.8 CU8: Recalibrate Photometry

```
CU8: Recalibrate photometry : Primary Scenarios

Precondition(s): Catalogue of photometric standards (primary, and/or
                 secondary) exists; DB system is read for updates. CU6 has
                 been run on calibration data.

Flow of events:

1. Join standard star observations on photometric nights to calibration
   data in SynopticSource table

2. Get photometric data for pairs of sources in all overlap regions taken
   on the the same nights as standard star data

3. Find best fit photometric solution to the above data.

4. Update model table with new solution and record previous solution.

5. Use overlaps to generate new ZPs for non-photometric nights

6. Update FITS headers files with new zeropoints

7. Update metadata table with new zeropoints and record previous zeropoints

8. Update magnitudes in detection and source tables.

Postcondition(s): DB system is left write-unlocked

Special requirement(s): implementation likely C/C++
                        application running on catalogue server
```

The algorithm for the photometric solution will be similar to the 'global photometric solution' algorithm employed for 2MASS [14], but will be flexible in allowing for nightly extinction solution (or common extinction measurements across several nights) as opposed to global extinction solutions. Implementation of step 3 will require a portable coding language that can easily handle large, sparsely filled matrices; this means that implementation will likely be in C/C++ as opposed to Python. Once again, Python scripting will be used to control the task workflow.

External updates to the zeropoints such as provided by CASU, or updates found through relative photometry measurements can be handled by running through steps 6 to 8 only.

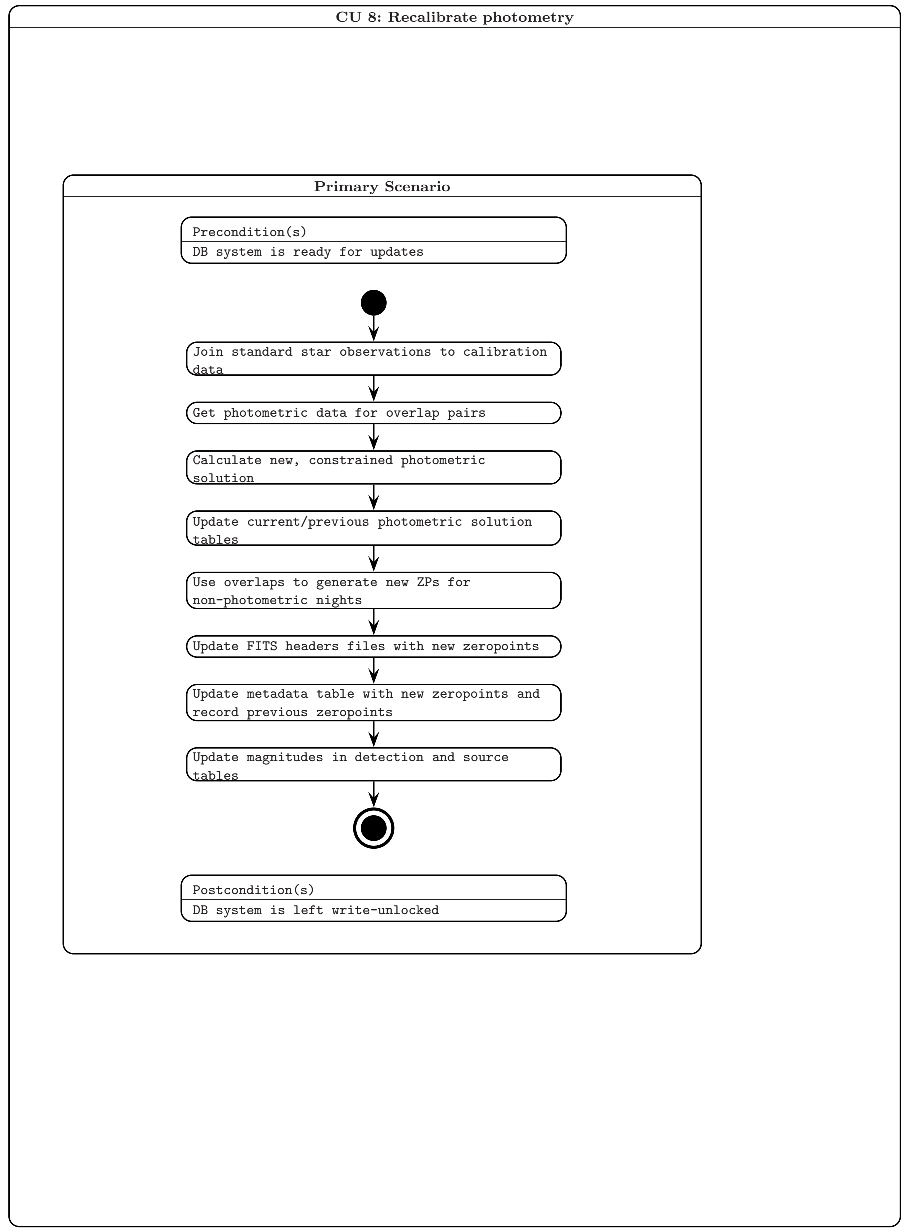

Figure 22: *Activity diagram for curation task CU8.*

8.9 CU9: Produce list measurements between passbands

```
CU9: Produce list measurements between passbands : Primary Scenarios

Precondition(s): DB system is ready for update

Flow of events:

1. For every field logged in merge event with one or more new images

   a) Extract field master source list from merged source list

   b) Pass master source list plus image details to CASU list-driven
      photometry tool

   c) Translate FITS binary output into intermediate file (native binary)

   d) Ingest intermediate file into list-driven remeasurements list

   e) Merge list remeasurements into sources

   f) Reseam the sources

   g) Reset all new/old flags for images in this field as now "old"

end

3. Update DB system curation log for CU9

Postcondition(s): DB system left write-unlocked

Special requirement(s):
```

Implementation details: this task will be implemented on the Linux server side. A Python script will query the appropriate merged source table on the DBMS server side via Python DBI to create a master driving source list for a given set of new images in the same field. The list, plus the location of the relevant images will be passed to the list–driven source re-measurement tool (a C application supplied by CASU). The FITS binary table output will then be translated and ingested using the same modules and methods as detailed in CU4.

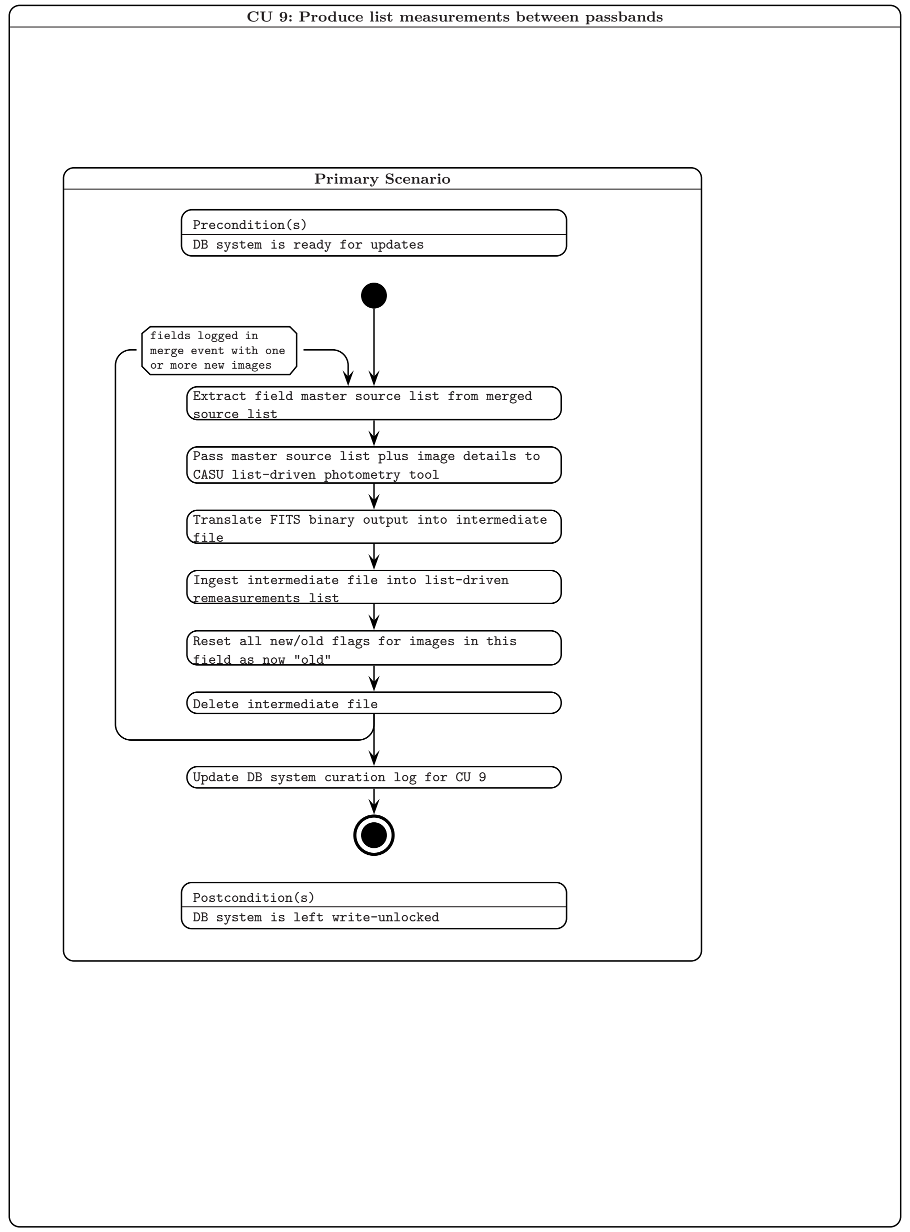

Figure 23: *Activity diagram for curation task CU9.*

8.10 CU10: Compute/update proper motions

```
CU10: Compute/update proper motions : Primary Scenarios

Precondition(s): Merged source catalogues have been created/updated;
                 DB system is ready for updates

Flow of events:

1. For every programme field having two or more images separated by at
   least one year, where more images than have previously been used are
   available

   a) Obtain detection lists for each independent image (i.e. no reruns)

   b) Associate detection lists (using existing merge information and
      proximity)

   c) Map out all position errors (small-scale) between measurement sets

   d) For each associated detection

      i) Compute full astrometric solution

      ii) Update astrometric attributes of corresponding merged source
          record

   end

end

3. Update DB system curation log for CU10

Postcondition(s): DB system is left write-unlocked

Special requirement(s):
```

Implementation details: algorithmically, this task is analogous to existing implementations for photographic plates within the SSA, where relative positional shifts are determined after positional error mapping has removed all systematic errors on all scales down to a few arcminutes [7]. It is presently unclear as to the best coding implementation for this task – large vector and matrix manipulation is required, which clearly rules out SQL and possibly Python.

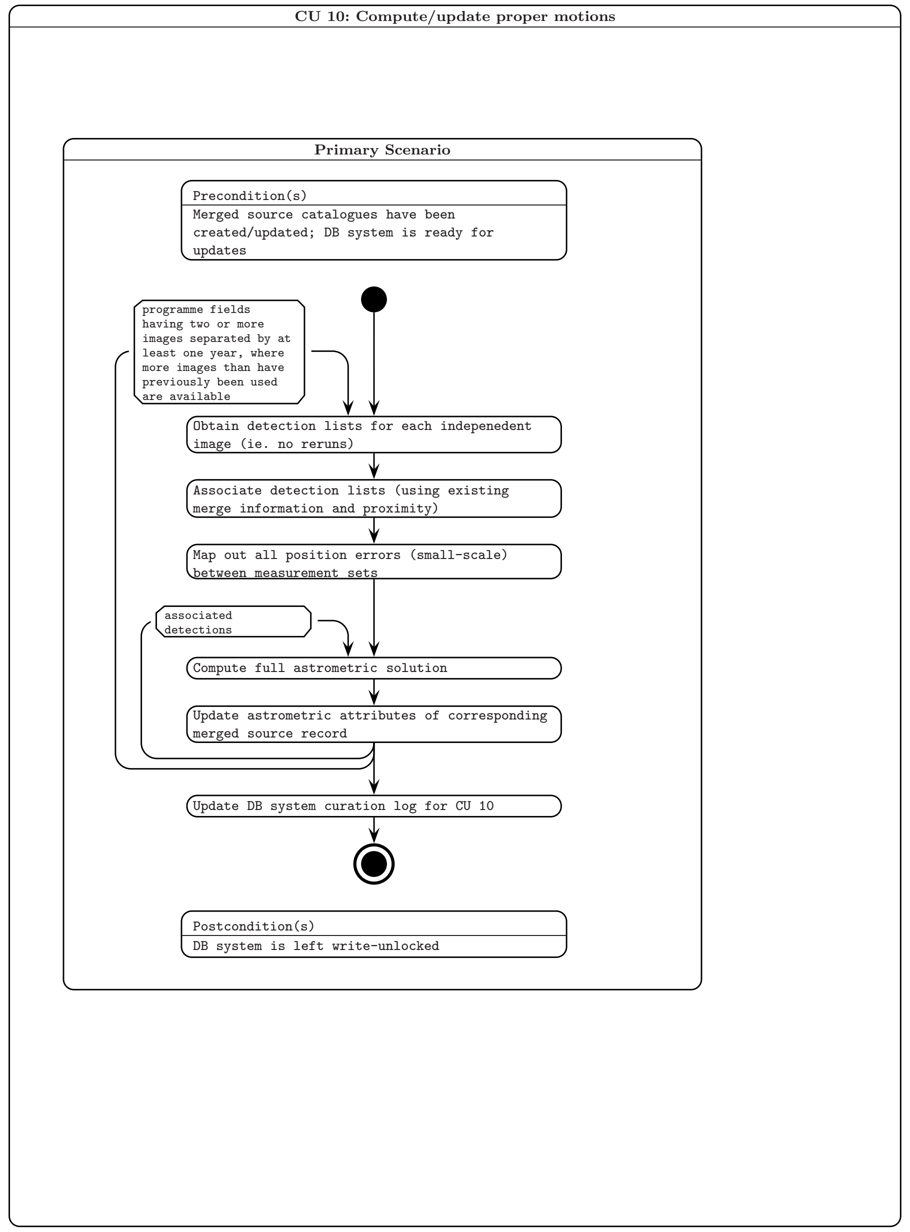

Figure 24: *Activity diagram for curation task CU10.*

8.11 CU11: Recalibrate astrometry

```
CU11: Recalibrate astrometry : Primary Scenarios

Precondition(s): DB system ready for update;
                 New astrometric reference catalogue exists and/or
                 low-level systematic errors are suspected to be
                 present

Flow of events:

1. Analyse data set(s) for systematic errors, creating "mask(s)"

2. If systematic errors are present OR new reference catalogue is
   available, then

   For each stored detector/combi frame;

      i) If new reference catalogue exists, then

         Move current astrometric calibration to previous;

         Derive new astrometric calibration

      endif

      ii) Update current astrometric calibration details, including any
          systematic error correction mask

      iii) Apply current astrometric calibration, to all records having
           explicitly stored co-ordinates

      iv) Update spatial index attributes

   end

3. Add new astrometric calibration version number

4. Recreate spatial indices for any subsequent curation procedure.

5. Update DB system curation log for CU11

Postcondition(s): DB system is left write-unlocked

Special requirement(s):
```

Implementation details: this task will not be executed frequently. An intercative sequence of steps using a combination of SQL scripts and some application codes will be employed.

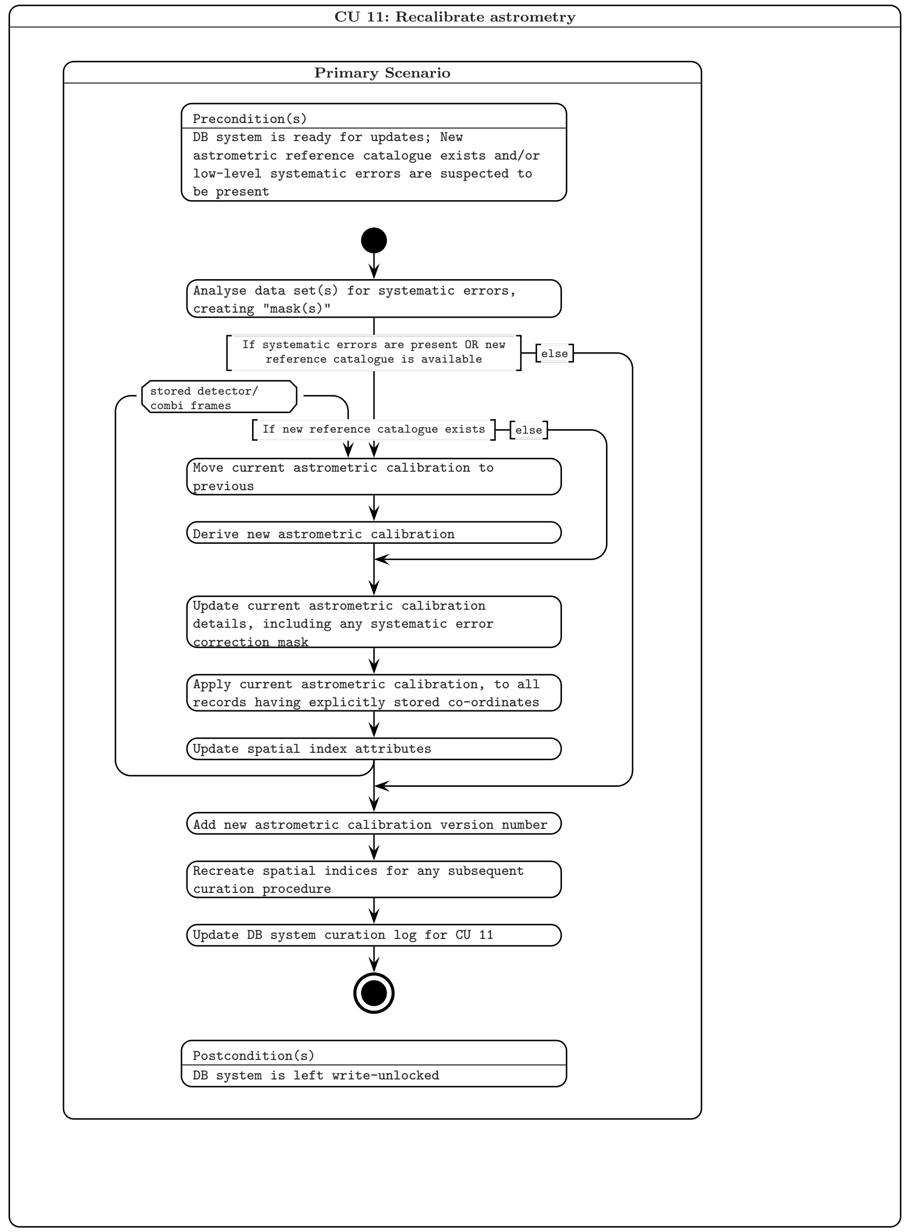

Figure 25: *Activity diagram for curation task CU11.*

8.12 CU12: Get external catalogues

```
CU12: Get publically released and/or consortium-supplied external
      catalogues : Primary Scenarios

Precondition(s): New data release is announced/communicated;
                 Stable catalogue data are available;
                 DB schema exists for the catalogue data;
                 DB system is ready for updates

Flow of events:

1. If previous version of dataset is held in the archive, then

      Remove previous version

   endif

2. Obtain catalogue product(s) (e.g.\ CD-ROM, DVD-ROM, or network copy)

3. If data not supplied as a DBMS database, then

      Translate catalogue product(s) into intermediate files

      Bulk load intermediate file(s) into DB system

      Remove intermediate files

   else

      Copy DB files into DB system

   endif

4. Update DB system curation log for CU12

Postcondition(s): DB system left write-unlocked

Special requirement(s):
```

Implementation details: this relatively infrequent task will require an interactive sequence of procedures. External catalogues other than the SDSS will likely be supplied in a variety of formats, hence small bespoke translation software modules will be written where necessary, on a case–by–case basis, for implementation on the Linux mass–storage server. Bulk loading scripts for execution on the DBMS server will be written in SQL to ingest intermediate translated files.

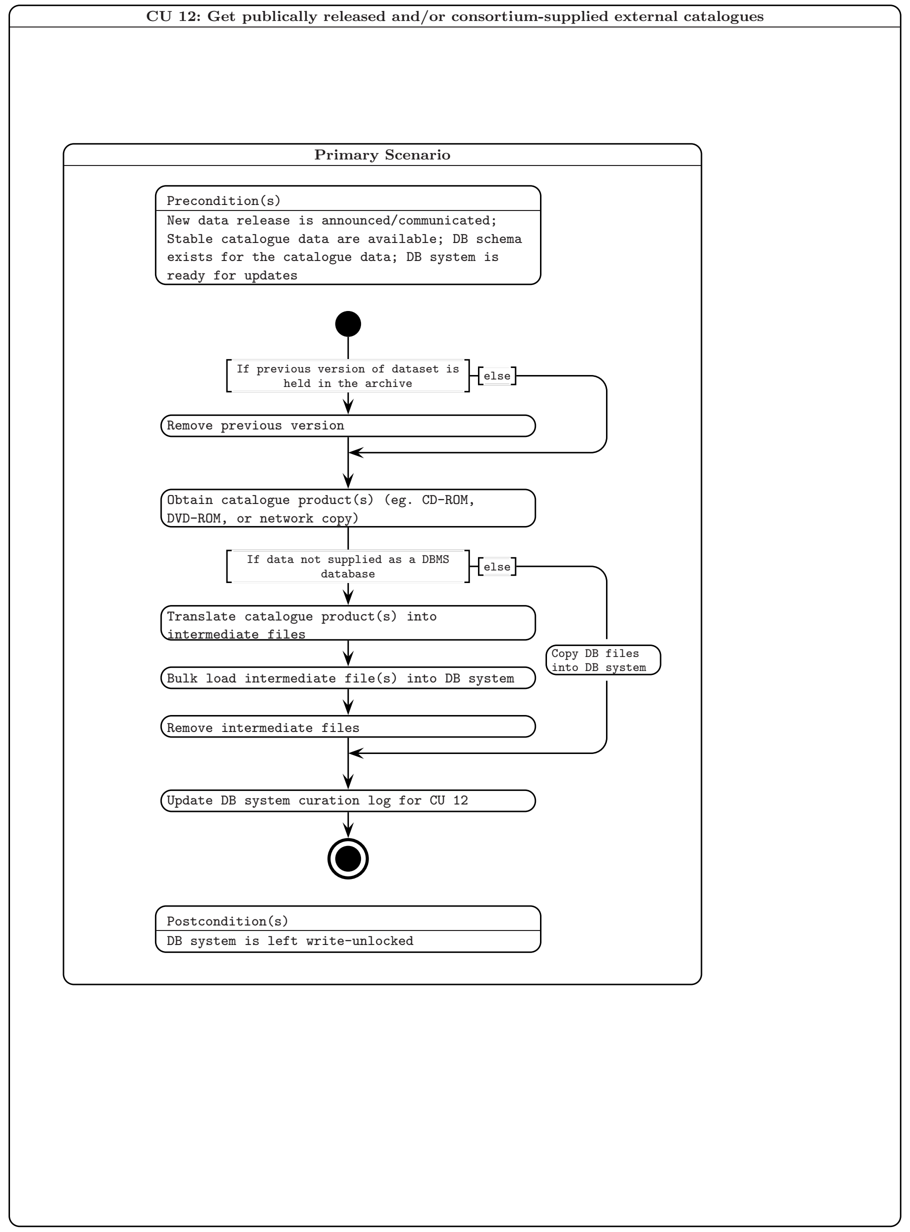

Figure 26: *Activity diagram for curation task CU12.*

8.13 CU13: Create library stacked /mosaiced image frame products

```
CU13: Create library stacked and/or mosaiced image frame
      products: Primary Scenarios

Precondition(s): DB system ready for update;
                 One or more new unstacked images exist for a given programme

Flow of events:


1. Select stacking or mosaicing option

2. Get list of required deep stacks or mosaics for given programme.

3. if mosaic:

     a) For each mosaic:
        i) Find already created deep stacks

        ii) Run CASU mosaicing tool to create mosaic

   elif stack:

     a) For each stack:

        i) Find available intermediate image and confidence frames that
           match the position and filter of the required stack.

        ii) For each intermediate stack:

           Check the data quality and remove poor frames

        iii) Use CASU stacking tool to produce deep stack or mosaic.

   endif
4. Check and fix image headers for each new product

end

5. Update curation log for relevant programme for CU13

Postcondition(s): DB system left write-unlocked
```

Implementation details: a Python script running on the Linux server side will query the DBMS via Python DBI to obtain lists of images to be manipulated. The image data and list obtained will then be passed to the appropriate combining tool (stacking or mosaicing tool supplied as C applications from CASU). Most stacking and mosaicing options are specified in RequiredStack or RequiredMosaic. One option in RequiredStack allows for a deep stack to be made up of a limited number (5) of intermediate frames, rather than all frames that pass the data quality control. In this case the 5 frames with the best seeing are selected.

A second mosaicing option is to convert an existing mosaic produced by SWARP to archive standard. This involves adding the correct headers, converting to multi-extension FITS (if necessary) and converting the weight image to a 16-bit integer confidence image with mean 100.

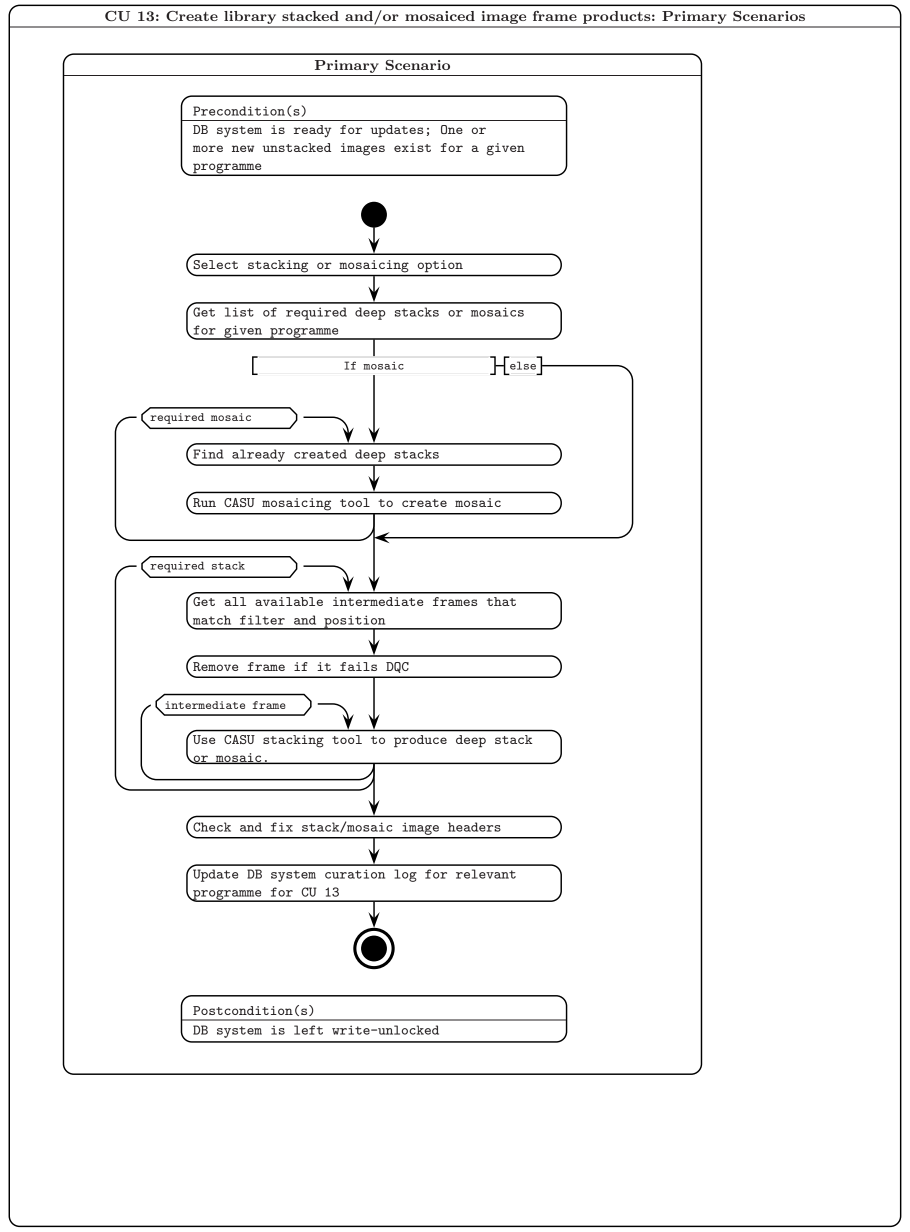

Figure 27: *Activity diagram for curation task CU13.*

8.14 CU14: Create standard source detection list from any new image frame product

```
CU14: Create standard source detection list from any new stacked/mosaiced
      image frame product: Primary Scenarios

Precondition(s): New stacked/mosaiced image exists;
                 DB system is ready for update

Flow of events:

1. Select the required stacks or mosaics from archive.

2. For each stack or mosaic

   a) Find the location of the latest production of the required
      stack/mosaic if any exists

   b) Produce catalogue FITS file using CASU source extraction tool or
      Source Extractor (whichever is required)

   c) Calculate the new zeropoint by comparing to intermediate stacks at
      the same location.

   d) Tidy up catalogue FITS headers

   e) Pass zeropoint and background information to image FITS headers

Postcondition(s): DB system left write-unlocked
```

Implementation details: a Python script will be implemented on the Linux server side that invokes the CASU–supplied standard source extraction tool (a C application) or Source Extractor for a specified image. The extraction options for each stack or mosaic are specified in RequiredStack or Required-Mosaic. Modules to strip the resulting image FITS keys and FITS binary table detection list (as for CU3 and CU4) will then execute, followed by DBMS ingest as detailed for CU4.

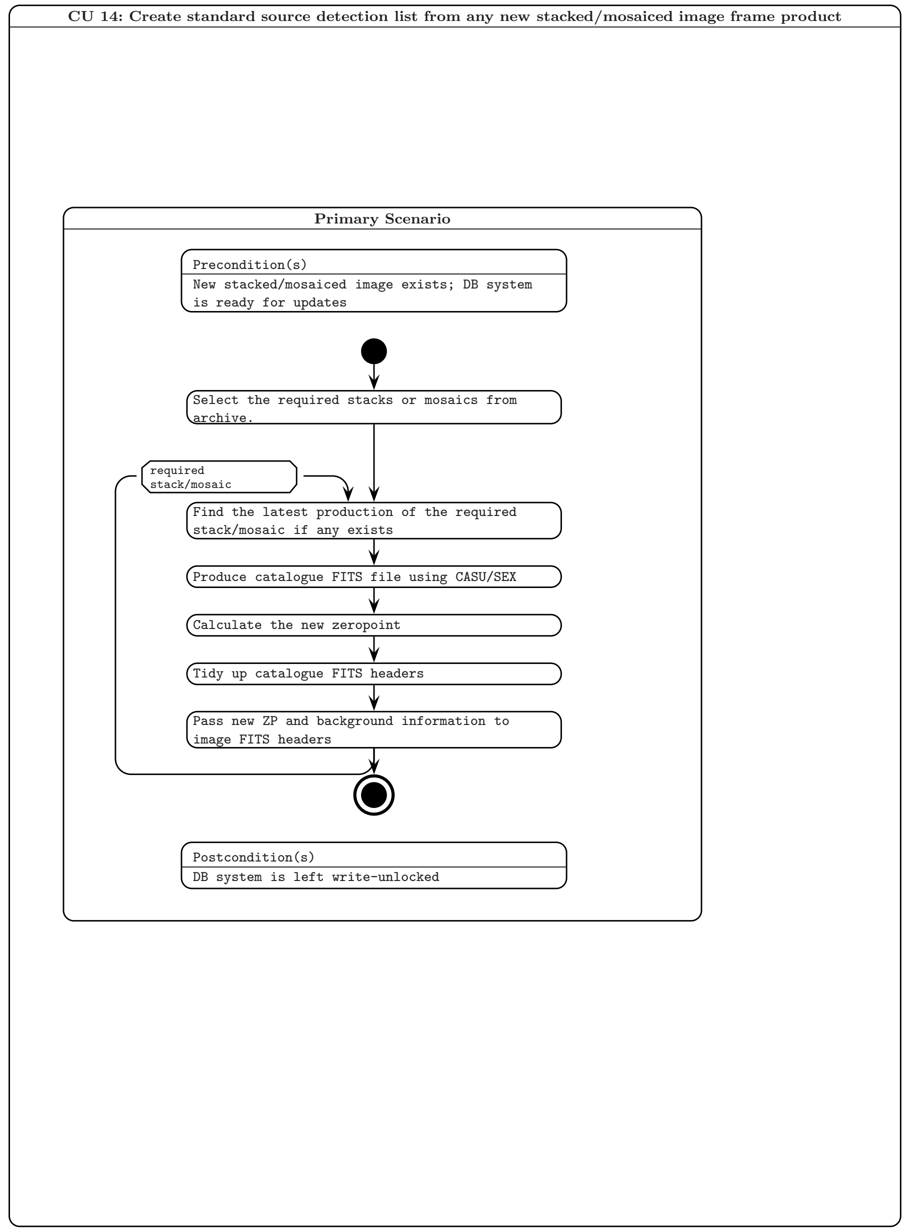

Figure 28: *Activity diagram for curation task CU14.*

8.15 CU15: Run periodic curation tasks for newly stacked images

```
CU15: Run periodic curation tasks: Primary Scenarios

Precondition(s): DB system ready for update;
                 CU14 has been run more recently than the last run of this
                 task

Flow of events:

1. if reqdDifferenceImage:

   a) Run CU5

   endif

2. Run CU7

3. Run CU9

4. Run CU16

5. Run CU17

6. Update curation log for CU15

Postcondition(s): DB system left write-unlocked
```

Implementation details: this task is sufficiently infrequently run that interactive invocation of tasks CU7, CU9, CU16 & CU17 will cover implementation; those scripts/applications will be implemented in a generic way to enable them to be applicable to this task. If difference images are required, as for instance for VPSP ELVIS, then CU5 must be run too.

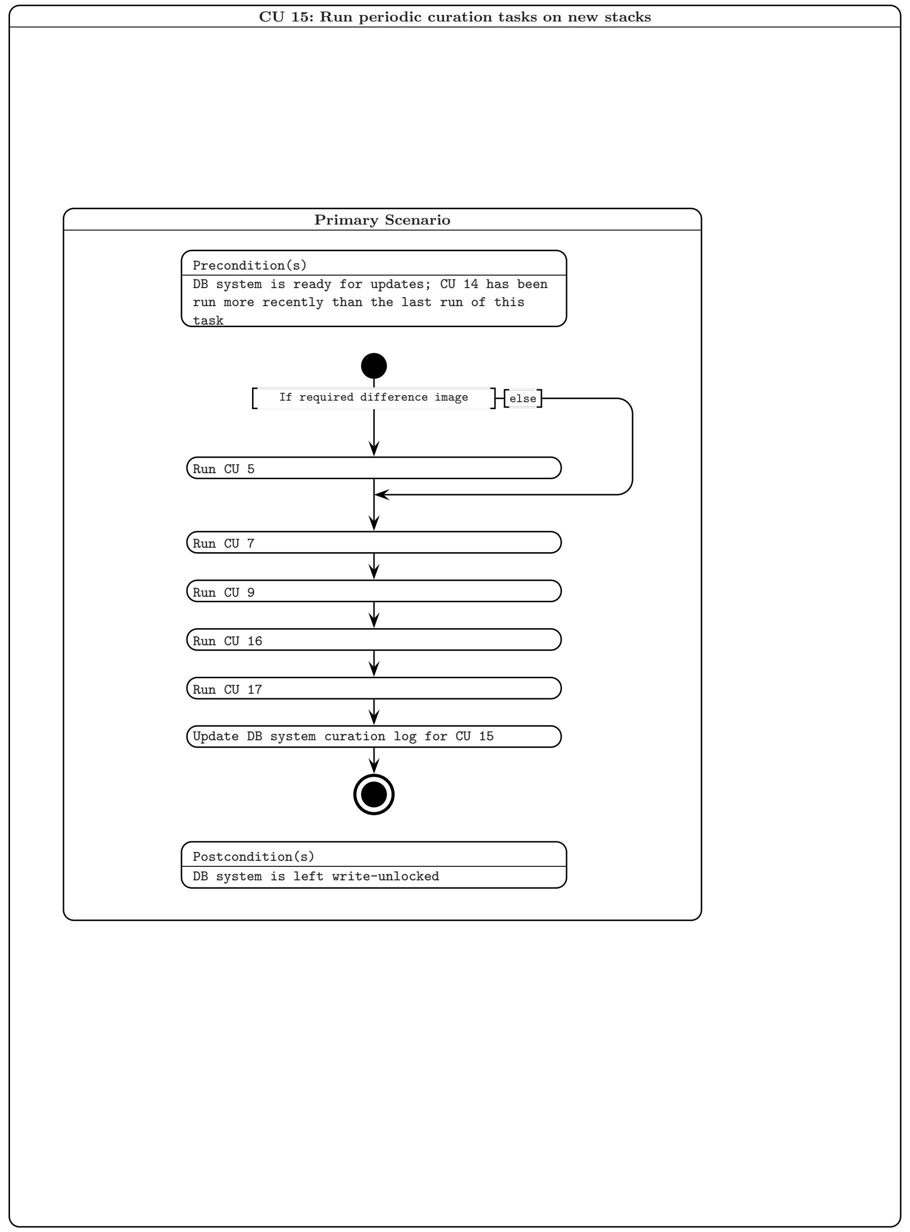

Figure 29: *Activity diagram for curation task CU15.*

8.16 CU16: Create default joins with external catalogues

```
CU16: Create default joins with external catalogues: Primary Scenarios

Precondition(s): DB system ready for update;
                 One or more new external catalogues exist

Flow of events:

1. Select required neightbour table(s) for the specified programme.

2. For each neighbour table

   a) Create and populate the Neighbour Table.

Postcondition(s): DB system left write-unlocked
```

Implementation details: spatial joins are impossible to achieve with requisite performance on the server side within the RDBMS. Bulk outgest of UIDs plus coordinates takes place prior to execution of a source matching application (see SAD [26]) that efficiently pairs lists of billions of rows. Finally, the bulk ingest of the neighbour table back into the DBMS completes this CU.

Both SourceNeighbour tables (objects in the survey Source table matched to other objects in the survey Source table) and cross-neighbour tables to external catalogues are created.

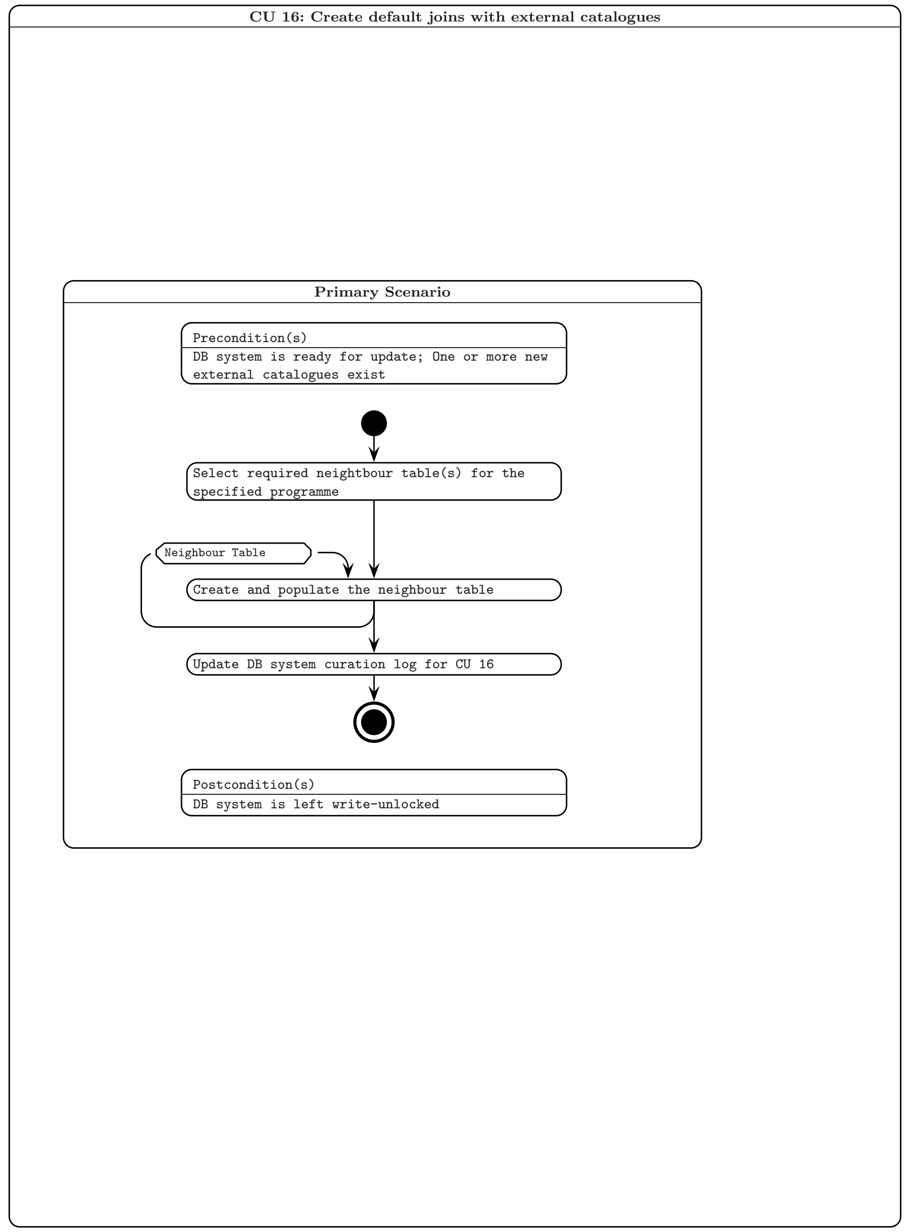

Figure 30: *Activity diagram for curation task CU16.*

8.17 CU17: Produce list–driven measurements between survey and external data

```
CU17: Produce list-driven measurements between survey detections and
      external imaging data: Primary Scenarios

Precondition(s):

Flow of events:

1. For every image set in external pixel dataset

   a) Determine sky coverage of image

   b) Extract list of survey sources that fall within this image, where no
      previous external measurements exist

   c) Pass this master driving list plus image details to CASU list-driven
      photometry tool

   d) Translate FITS binary output into intermediate file (native binary)

   e) Ingest intermediate file into appropriate list-driven measurement
      table

   f) Delete intermediate file

end

3. Update curation log for CU17 for relevant programme

Postcondition(s): DB system left write-unlocked

Special requirements:
```

Implementation details: this task will run as a Python script on the Linux server side, where the list–driven source re-measurement tool will be supplied by CASU as a C application. Generally, implementation will follow that for CU9.

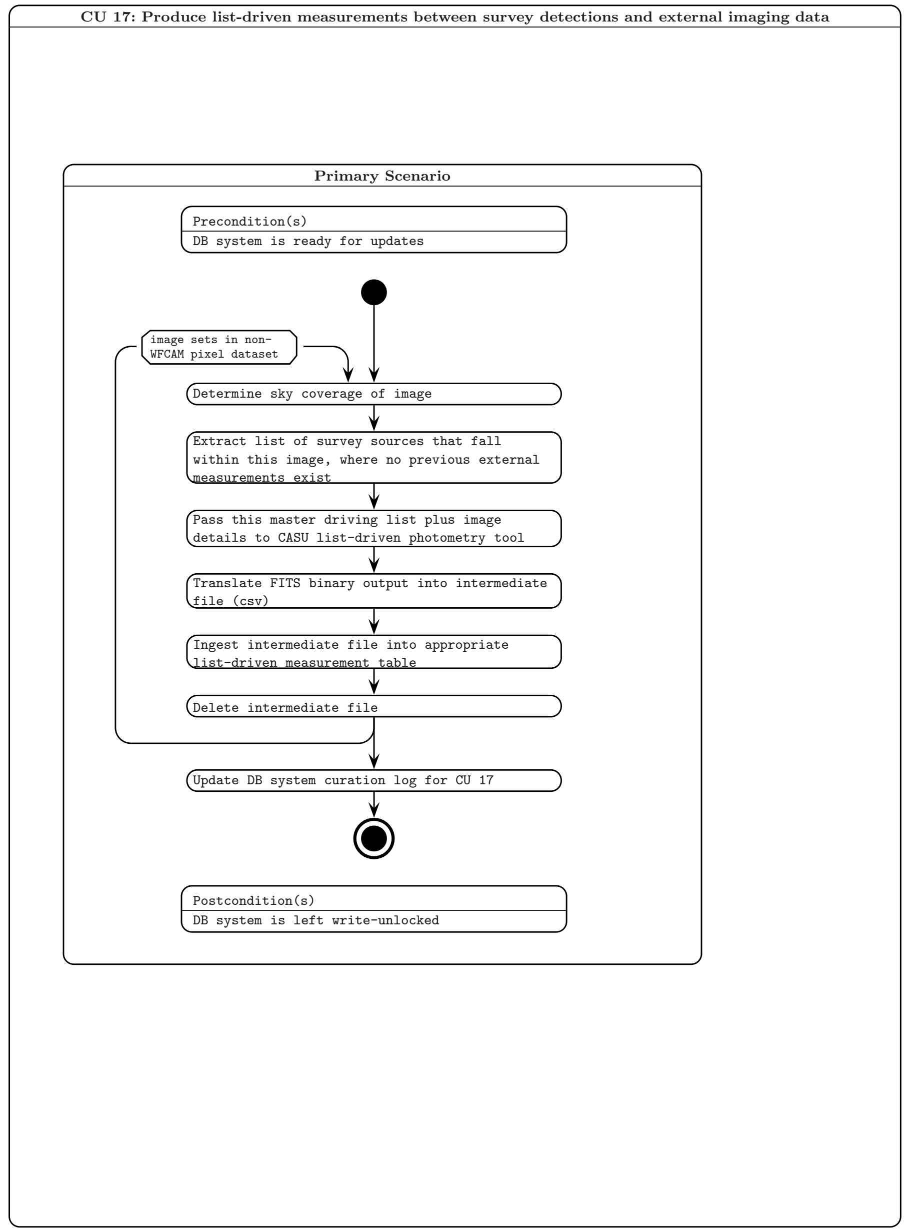

Figure 31: *Activity diagram for curation task CU17.*

8.18 CU18: Create/recreate table indices

```
CU18: Create/recreate table indices: Primary Scenarios

Precondition(s): DB system is ready for update

Flow of events:

1. For every programme that has been updated since the last index update

   a) For every table for that programme

      i) Create/recreate required indices for this table

   end

   b) Update curation log for this programme

end


Postcondition(s): DB system left write-unlocked
```

Implementation details: this task will be implemented using SQL scripts on the DBMS server side.

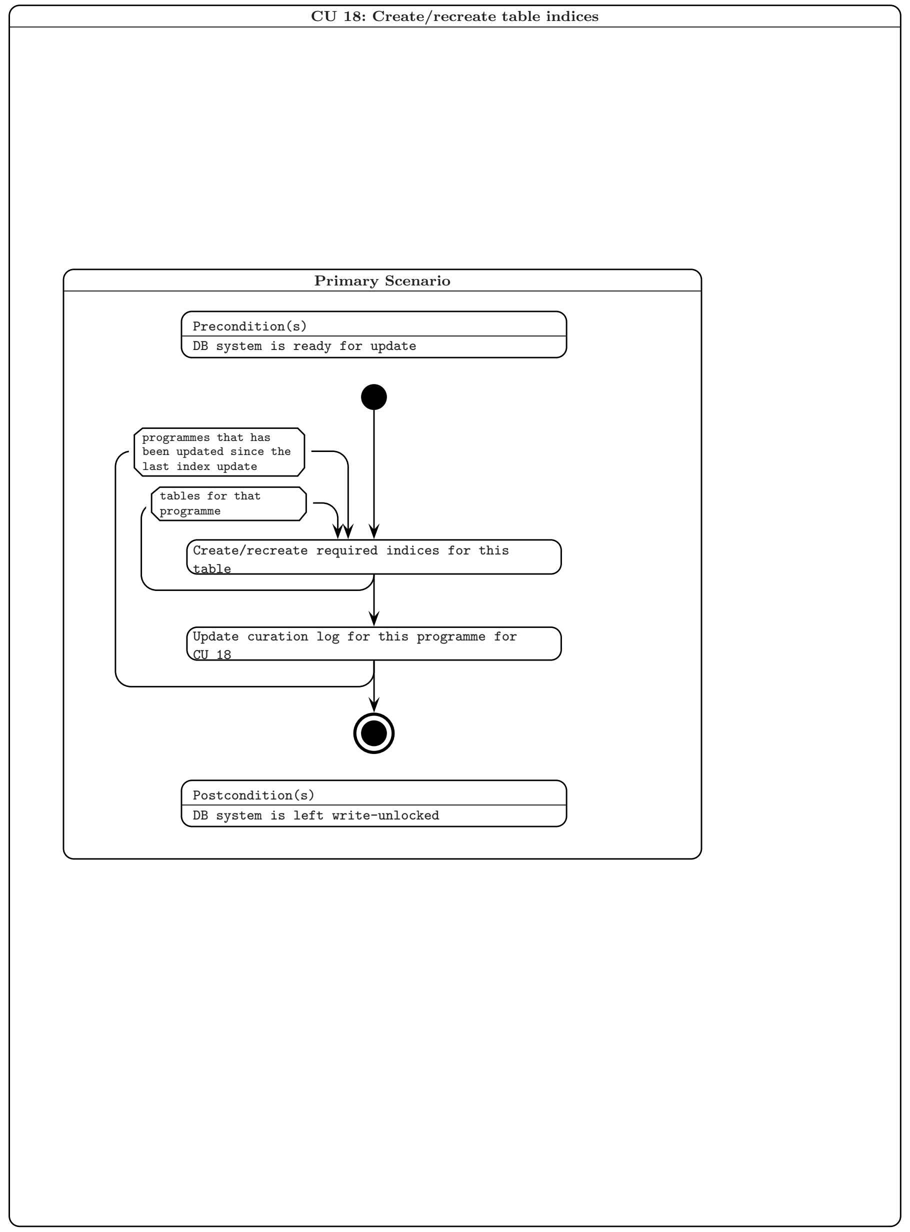

Figure 32: *Activity diagram for curation task CU18.*

8.19 CU19: Verify, freeze and backup

```
CU19: Verify, freeze and backup: Primary Scenarios

Precondition(s): DB system is ready for update

Flow of events:

1. Verify that all the programmes are ready and log the release

2. If non-survey release

   a) Copy out the data

3. else if survey release

   a) Freeze and backup the VSA.

   b) Restore the backup as release name

   c) Expunge unwanted data and compact to release unused space

   d) Set DB options for release

Postcondition(s): DB system left write-unlocked
```

Implementation details: Python script on the Linux server side, which will control SQL scripts.

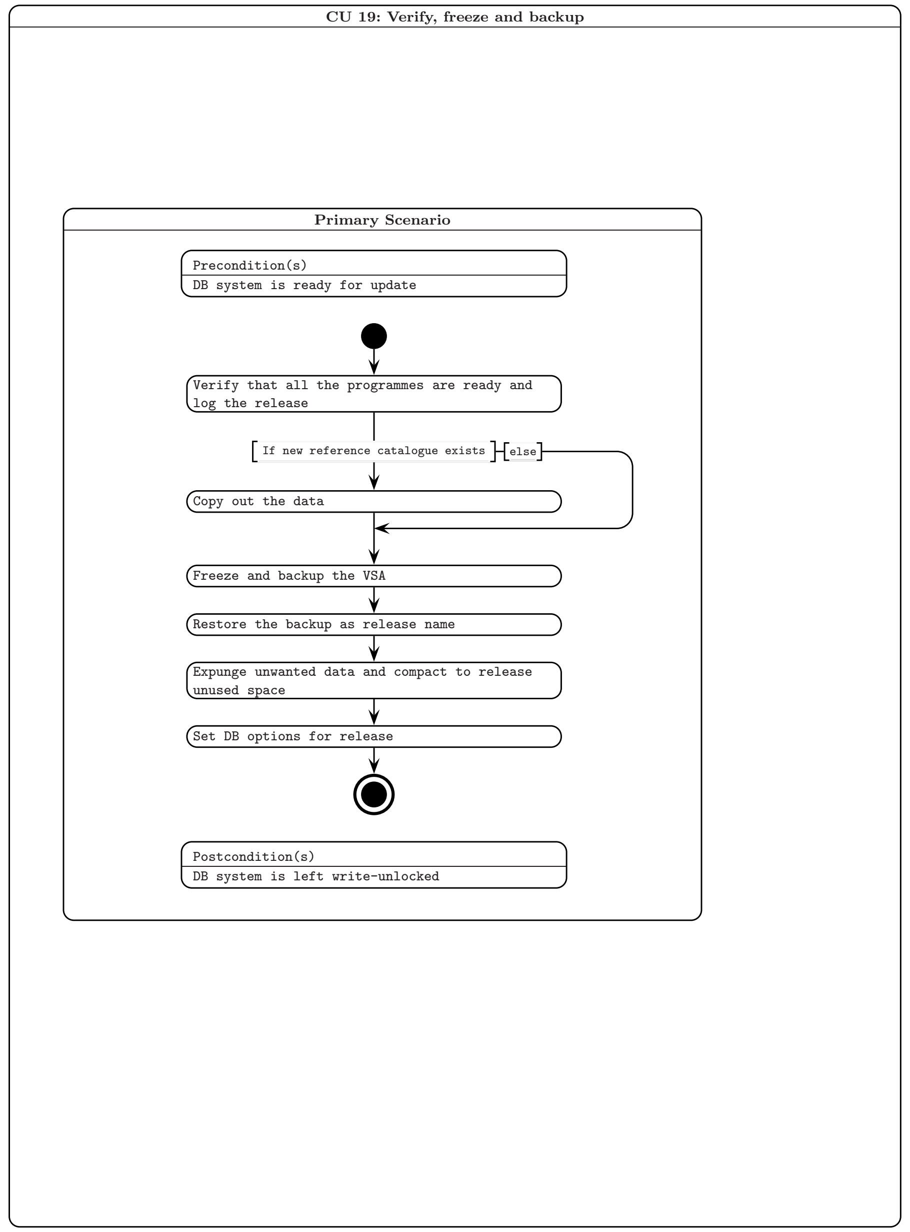

Figure 33: *Activity diagram for curation task CU19.*

8.20 CU20: Release – place online new DB product(s)

```
CU20: Release - place online new DB product(s): Primary Scenarios

Precondition(s): Public access to online server is disabled

Flow of events:

1. Remove the current "world" view

2. For every DB being newly released

   a) Remove previous corresponding existing DB

3. Recreate "world" view on all world survey subsets

4. Update release information in both online and offline curation log

5. Update curation log for CU20

Postcondition(s): Public access to online server enabled

Special requirements:
```

Implementation details: SQL script is used to implement this task. The script runs on the public–access server system.

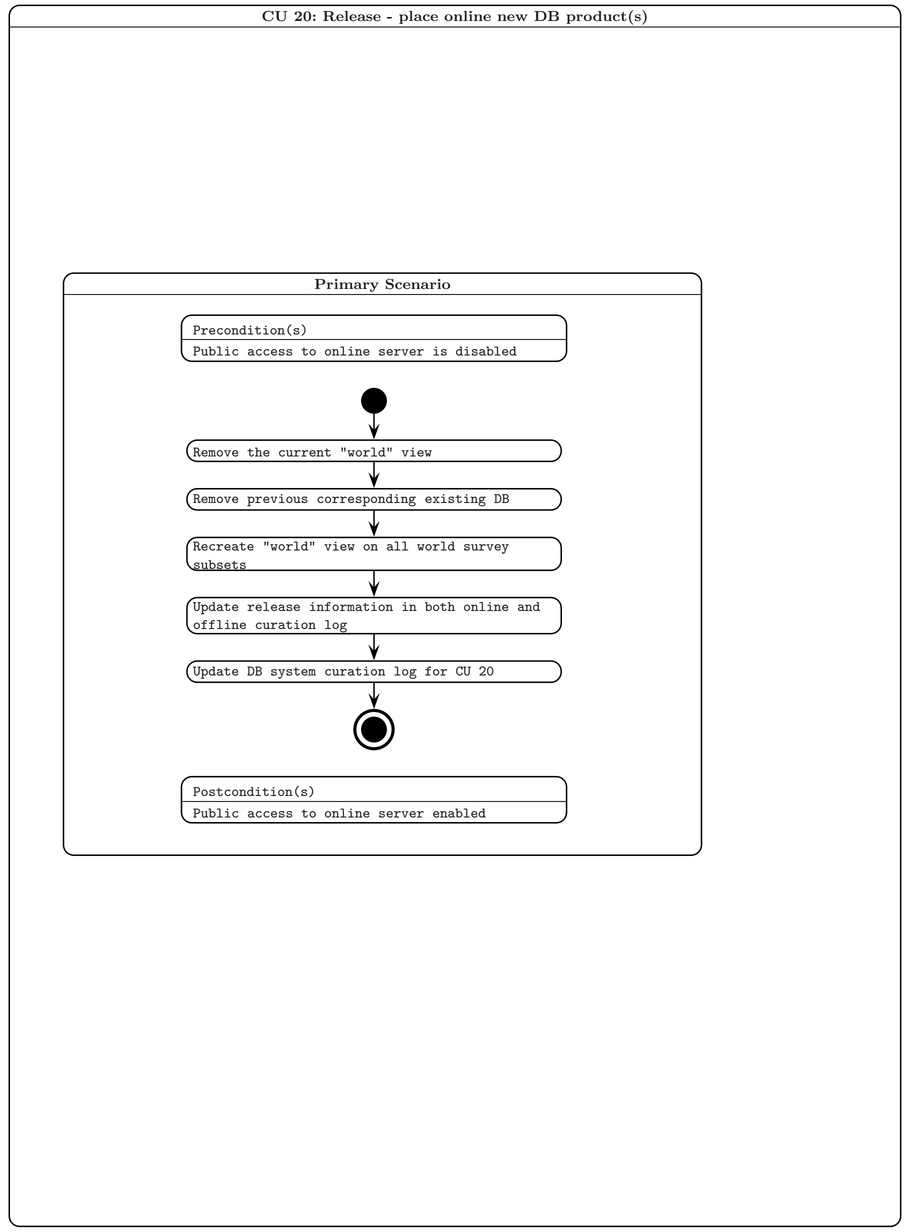

Figure 34: *Activity diagram for curation task CU20.*

8.21 CU21: Create non-survey products

```
CU21: Create non-survey products

Flow of events:

1. Create SQL setup schema and programme schemas

2. Copy over the schema files to the file share on the load server and run
   the scripts

3. Create the programme list for cu4 and check it in.

4. Ingest any available data for these programmes

Special requirements:
```

References

[1] Hierarchical Triangular Mesh indexing; http://www.sdss.jhu.edu/htm/index.html

[2] The Sloan Digital Sky Survey at Johns Hopkins University; http://www.sdss.jhu.edu/

[3] Jim Gray, private communication.

[4] 20 Queries for the SuperCOSMOS Science Archive http://www.roe.ac.uk/~nch/wfcam/misc/20queries.sql

[5] SQL Server database schema for the SuperCOSMOS Science Archive http://www.roe.ac.uk/~nch/wfcam/misc/sss1412.sql

[6] Multi–threaded query agent and engine for a very large astronomical database, Thakar, A. et al., in Proc. ADASS IX, eds. Manset, Veillet & Crabtree, ASP Conf. Ser., 216, 231 (2000)

[7] The SuperCOSMOS Sky Survey, Paper III: Astrometry, Hambly et al., MNRAS, 326, 1315 (2001)

[8] Photometric Calibration of UKIDSS data, Hodgkin, S. et al., in preparation, (2006)

[9] Photometric Recalibration Design Document http://www.roe.ac.uk/~njc/VDF-WFA-WSA-phot.ps.

[10] The UKIRT Infrared Deep Sky Survey Early Data Release, Dye, S. et al., MNRAS, accepted (2006)

[11] The UKIRT Infrared Deep Sky Survey Data Release One, Warren, S. et al., MNRAS, in preparation (2006)

[12] WFCAM Science Archive http://surveys.roe.ac.uk/wsa/pre/index.html

[13] Hierarchical Equal Area isoLatitude PIXelisation of the celestial sphere; http://www.eso.org/science/healpix/index.html

[14] A global photometric analysis of 2MASS calibration data, Nikolaev et al., AJ, 120, 3340 (2000)

[15] Achieving 1% photometric accuracy with the ESO wide field imager, Manfroid et al., ESO Messenger, June 2001, 16

[16] Archive scientist curation use cases for the WSA; http://www.roe.ac.uk/~nch/wfcam/curate/curate.html

[17] The SuperCOSMOS Sky Survey; http://www-wfau.roe.ac.uk/sss

[18] The SuperCOSMOS Halpha Survey; http://www-wfau.roe.ac.uk/sss/halpha

[19] Personal SkyServer; http://research.microsoft.com/~gray/SDSS/personal_skyserver.htm

[20] The WFCAM data processing pipeline, Irwin, M. et al., in preparation (2006)

[21] Real–time difference imaging analysis of MOA Galactic Bulge observations during 2000, Bond I. et al., MNRAS, 327, 868 (2001)

[22] VISTA Science Archive Science Requirements Analysis Document; http://www.roe.ac.uk/~nch/wfcam/srd/wsasrd/wsasrd.html

[23] VISTA Science Archive interface control document;
http://www.roe.ac.uk/~nch/wfcam/VDF-WFA-VSA-004-I1/VDF-WFA-VSA-004-I1.html

[24] VISTA Science Archive data flow document;
http://www.roe.ac.uk/~nch/wfcam/VDF-WFA-VSA-005-I1/VDF-WFA-VSA-005-I1.html

[25] VISTA Science Archive hardware/OS/DBMS design document;
http://www.roe.ac.uk/~nch/wfcam/VDF-WFA-VSA-006-I1/VDF-WFA-VSA-006-I1.html

[26] VISTA Science Archive Software Architecture Design;
http://www.roe.ac.uk/~nch/wfcam/VDF-WFA-VSA-009-I1/VDF-WFA-VSA-009-I1.html

[27] VISTA Science Archive management and planning document;
http://www.roe.ac.uk/~nch/wfcam/VDF-WFA-VSA-003-I1/VDF-WFA-VSA-003-I1.html

[28] The UKIDSS Proposal; http://www.ukidss.org/sciencecase/sciencecase.html

[29] Example SQL schema scripts for the WFCAM Science Archive
http://www.roe.ac.uk/~nch/wfcam/misc/wsa.sql

[30] Examples of WSA usage Usages of the WFCAM Science Archive

9 APPENDICES

9.1 Curation use cases

For the purposes of progressing database design in the WSA/VSA project, this document describes a set of 'curation use cases'. The idea is to examine the VSA Science Requirements Document and Usages of the WFCAM Science Archive to define the required curation procedures, and hence to broadly outline the data entities in the VSA and their upkeep. In this context, 'curation' refers to the process of transfering and ingesting VISTA data, generation of new data products from the standard pipeline data products, and management of those data products in the Science Archive.

The curation use cases are split into four broad categories, both for reasons of clarity and because requirements on various timescales give rise to distinct tasks:

- Daily curation use cases
- Periodic (weekly/monthly) curation use cases
- Occasional curation use cases
- Release time curation use cases

Also, the idea is to spread curation tasks in time as much as possible so as to spread the processing load on catalogue servers.

Because various survey data have various proprietory periods, it will be necessary to curate various databases having different access restrictions. We propose to curate online ESO, open time and 'world' (i.e. unrestricted access) databases, as well as incremental, offline versions of the same.

A database entity (table) will be required that keeps track of which curation tasks have been applied at any given time and to any given programme dataset.

9.1.1 Daily curation use cases

The following tasks occur on a daily basis. Pipeline processing will take place on a night–by–night basis and daily transfer and ingest is necessary to keep up with the end–to–end system data rate:

CU1: Obtain science data from CASU.

Pipeline processed data will be transfered from the pipeline processing centre (CASU) via the internet. This use case consists of

- transfering data
- verfication of transfered data - ensure integrity of data files obtained
- log transfer - keep track of which data have been transfered and at what time

Logged information from all curation tasks must be held in the database so that it is queryable by users. These logs will take the form of separate tables of data, or in this case, logged information held with the image metadata.

CU2: Create 'library' compressed image frame products

For speedy access to image data (e.g. for finder chart purposes), compressed images are to be employed:

- JPEGs

- include North and East axes.

CU3: Ingest details of transfered and library compressed images into archive DBMS, including their standard astrometric and photometric calibrations

Gbyte–sized FITS images will not be stored as BLOBs in the DBMS as this would result in heavy IO usage that would impact on more time–critical catalogue DBMS queries. Images will be stored as flat files; however their details (FITS header keywords, filenames, astrometric details, pipeline processing information, quality control information etc.) will be stored in the DBMS to track these files (from curation and use points of view).

CU4: Ingest single frame source detections into appropriate detection lists in the incremental (offline) archive DBMS.

All output from the standard pipeline source extraction algorithm needs to be stored in the archive.

9.1.2 Periodic (weekly/monthly) curation use cases

The following use cases occur periodically. The idea here is that the VSA usages require production of data products (catalogues) within the DBMS - it will be impossible merely to append data to existing entities (tables) within the DBMS since creating a science-usable survey dataset requires many non-linear operations (e.g. pairing, indexing, astrometric and photometric recalibration). Exactly what timescale is practical will become clear with experience (e.g. weekly, fortnightly or monthly).

CU5: Create "library" difference image frame products

After accumulation of a certain amount of image data it will be possible to update products resulting from combinatorial operations on individual image frames where this is not possible on a daily basis if image multiples are not guaranteed to be observed and processed together (generally the case).

CU6: Create Synoptic Source Tables

Standard CASU processing will not produce any merged catalogue products. Once the single passband detections are stored within the archive, a small number of the best individual observations will be stacked and a catalogue extraction code run to generate a master catalogue. A neighbour table will be created between this source list and the detection table. For each object in the source list variability statistic(s) will be calculated so that scientists can easily find interesting variable or non-variable objects. A SynopticSource table with colour information at mulitple times can also be generated. This will use tasks from curation usecases 13, 14, 7, 8, 9 and 16. The master catalogue only has to be created once, and additional observations can be added in easily. This usecase will be used with all synoptic surveys such as the proposed VVV and VMC surveys as well as the WFCAM Planet Transit Survey and all the calibration data taken by UKIRT and VISTA.

CU7: Create/update merged source catalogues to the prescription available for a given survey from the appropriate detection list

Standard CASU processing will not produce any merged catalogue products. Once the single passband detections are stored within the archive, a detection association algorithm will execute and produce merged multi–colour, multi–epoch records appropriate to the available data within a given survey dataset. Separate merged source catalogue products will be required for different sub-surveys and open time programmes. Each merged catalogue dataset will have an associated table of logged information containing details of the merge run and what fields have been included to date.

CU9: Produce list–driven measurements between passbands

Standard source detection will involve setting a threshold for detection; however in the context of data mining it may be important to have source extraction and detection limits (i.e. photometric measurements) at positions and with apertures/profiles/deblending defined, for example, by detections across all bands. This philosophy follows the SDSS, where flux measurements at standard positions and in standard apertures are made in all bands when a detection is present in at least one band.

CU10: Compute/update proper motions and other multi–passband derived quantities

Other multi–colour attributes include, for example, extinction and dereddened apparent magnitudes which have been suggested within the UKIDSS GPS and would apply in the VPSP VGPS.

9.1.3 Occasional curation use cases

Occasional tasks are associated with newly available, externally produced data products from other survey programmes that are required to be held in the VSA for the purposes of joint querying to enable many of the science goals of the VPSP, for example. Astrometric and photometric recalibration will also be undertaken occasionally.

CU8: Recalibrate photometry

Full–blown photometric solutions over many photometric nights within an observing block will be undertaken. This will consist of:

- Associating detection records with photometric standards
- Merging sources in regions of overlap.
- Solving for photometric coefficients (zeropoints, first– and second–order extinction coefficients, etc.)
- Updating calibration coefficients within the DB, and updating zeropoints in the FITS files, the metadata tables and magnitudes and colours in all the detection, source and list driven tables.
- Previous zeropoints and calibration coefficients must be logged and timestamped.

If external calibration occurs then the undating of zeropoints and magnitudes and the logging of the previous zeropoints can be done separately.

CU11: Recalibrate image/detection astrometry

After data have accumulated for a sufficient time, low-level systematic errors in astrometry may become apparent, and it may be possible to remove these; furthermore, new astrometric reference catalogues may become available over the lifetime of the VSA, in which case astrometric recalibration is in order. If this is done, the spatial indexes will need to be undated.

The simplest usages of the VSA (position/proximity searches) and also curation use cases such as pairing will be made much more efficient if the database entities are spatially indexed in some way (in it's simplest form, such indexing would take the form of sorting on one co-ordinate; the VSA will use a more sophisticated approach, e.g. Hierarchical Triangular Mesh(HTM) indexing.

CU12: Get publicly released and/or consortium–supplied (e.g. complementary optical) external catalogues

It will be necessary to update the locally stored (but externally produced) survey products (e.g. SDSS, 2MASS, etc.) where new releases of those products have been made.

9.1.4 Release curation use cases

Proprietory PI programmes should be released to the proprietors asap; updates to ESO Public surveys will naturally occur on a timescale dictated by pre-arranged release dates.

CU13: Create "library" stacked and/or mosaiced image frame products

After accumulation of a certain amount of image data it will be possible to update products resulting from combinatorial operations on individual image frames. This will be necessary for programmes such as the VISTA-UKIRT Ultra Deep Survey, ULTRA-VISTA and ELVIS.

CU14: Create standard source detection list from any new "library" frame product(s)

For example, if a new deep stack or mosaic has been made, then standard source detection should be run on the resulting image data.

CU15: run periodic curation tasks 7, 9, 16 & 17

– for any newly created stacked/mosaiced image product. In the case of deep emission line programmes like the proposed ELVIS survey, it may also be necessary to run CU5 too.

CU16: Create default joins with external catalogues

For the purposes of cross–IDs/neighbours for rapid cross-querying.

CU17: Produce list-driven measurements between VISTA detections and non–VISTA image data, where possible

Again, following CU9, and with usages such as U1 [30] in mind. That example concerns the UKIDSS LAS and SDSS UGRIZ imaging data, where IR sources with upper limits in IZ are sought for candidate very cool objects. The same type of data could also be used to identify extremely high redshift QSOs.

CU18: Create/recreate table indices

Within a given table, an index will be created on a combination of commonly used attributes so that at query time, the query optimizer will make use of these indices to greatly enhance performance.

CU19: Verify, 'freeze', and backup

Verification takes the form of examining the curation log to check that no further curation is needed for any given programme/survey dataset. This curation task will create a 'world' readable subset of the given programme dataset based on the proprietory period of the observations (tracked through the database) and the current release date. Prepared survey DBs should be fixed and backed up for security

CU20: Release – place new DB products online

This task is the final step: any newly created database products will be placed on a publicly accessible catalogue server. At the same time, a 'world'–readable 'view' of the programme subsets will be created to present a single, logical database of all VISTA observations having unrestricted access at that date.

9.1.5 Non Survey Data

CU21: Make new non-Survey dataset

This sets up the schema and creates all the required tables for a new non-survey data set. Additionally it will ingest and available data into the catalogue tables - the image data will already have been ingested into Multiframe and MultiframeDetector.

9.2 The WFCAM Science Archive database

The WFCAM Science Archive (WSA) has been developed as a prototype multi Tbyte archive to gain experience prior to arrival of large amounts of VISTA data. The WSA is an implementation of the UKIRT Deep Sky Survey (UKIDSS; [28]) in Windows/SQL Server. The WSA is the work up (Phases 1, 2 & 3) to the VSA (Phase 4):

- spatial indexing is required for speedy access;
- many photometric, astrometric and morphological descriptors per detected source;
- multi–colour, multi–epoch merged source catalogue results from multi–passband, multi–epoch imaging;
- deep data built up from observations over many nights;
- synoptic data taken to measure variability;
- joins to external catalogues providing data in the optical (SDSS, SSA, MGC), xray (ROSAT, XMM), far-IR (IRAS, GLIMPSE), radio (FIRST) and spectroscopic data (SDSS, MGC) to compliment the near-IR data.

The synoptic data requirement was not originally part of the requirements for the WSA, but the Planet Transit Survey (UK PATT U/06A/52) and the calibration data have prompted work on CU6, although this usecase is still in a preliminary state.

The pixel size of VISTA is more closely matched to the Nyquist sampling for the expected seeing at the Paranal site than the WFCAM pixel size is to the seeing at Mauna Kea, so interleaving will not be such a priority with VISTA data.

The development of the WSA, with increasing amounts of WFCAM data ($\sim$ 1.5 years, $\sim$ 25Tb of pixel data and $\sim$ 1Tb) is an important test case for VISTA. The nightly data rate is $\sim$ 100Gb and VISTA will be about 2-3 times this, but unlike WFCAM which is one instrument of many on UKIRT and is only on the telescope for $\sim$ 60% of the time, the infra-red survey camera will be the only instrument on the VISTA telescope and will be used on every clear night apart from a small fraction of the year when it is being serviced. The total volume of data from VISTA will be $\sim 3-5$ times that of WFCAM.

9.3 Example SQL script files for multiframe and associated calibration table schemas

```
CREATE TABLE vhsDetection(
-------------------------------------------------------------------------------
--/H Contains the individual VHS source detections from Multiframe.
--
--/T This table contains the individual detections for sources originating
--/T from multiframe images taken for the UKIDSS Large Area Survey (VHS).
--/T
--/T Required constraints: primary key is (multiframeID,extNum,seqNum)
--/T                       (multiframeID,extNum) references MultiframeDetector(multiframeID,extNum)
-------------------------------------------------------------------------------
multiframeID    bigint not null,    --/D the UID of the relevant multiframe  --/C ID_FRAME
extNum          tinyint not null,   --/D the extension number of this frame  --/C NUMBER
```

```
cuEventID       int not null,       --/D UID of curation event giving rise to this record  --/C REFER_CODE
seqNum          int not null,       --/D the running number of this detection  --/C ID_NUMBER  --/F Sequence_number  --/I The running number for ease of reference, in strict order of image detections.
filterID        tinyint not null,   --/D UID of combined filter (assigned in WSA: 1=Z,2=Y,3=J,4=H,5=K,6=H2,7=Br,8=blank)  --/C INST_FILTER_CODE
--
--  CASU standard source detection attributes:
--
isoFlux         real not null,      --/D Instrumental isophotal flux counts  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Isophotal_flux  --/G allDetection::isoFlux
isoMag          real not null,      --/D Calibrated isophotal magnitude  --/U mag  --/C PHOT_INT-MAG  --/Q isoFlux
x               real not null,      --/D X coordinate of detection  --/U pixels  --/C POS_PLATE_X  --/F X_coordinate  --/I Intensity-weighted isophotal centre-of-gravity in X.
xErr            real not null,      --/D Error in X coordinate  --/U pixels  --/C ERROR  --/F X_coordinate_err  --/I Estimate of centroid error.
y               real not null,      --/D Y coordinate of detection  --/U pixels  --/C POS_PLATE_Y  --/F Y_coordinate  --/I Intensity-weighted isophotal centre-of-gravity in Y.
yErr            real not null,      --/D Error in Y coordinate  --/U pixels  --/C ERROR  --/F Y_coordinate_err  --/I Estimate of centroid error.
gauSig          real not null,      --/D RMS of axes of ellipse fit  --/U pixels  --/C MORPH_PARAM  --/F Gaussian_sigma  --/G allDetection::gauSig
ell             real not null,      --/D 1-b/a, where a/b=semi-major/minor axes  --/C PHYS_ELLIPTICITY  --/F Ellipticity
pa              real not null,      --/D ellipse fit orientation to x axis  --/U degrees  --/C POS_POS-ANG  --/F Position_angle  --/I Angle of ellipse major axis wrt x axis.
aProf1          int not null,       --/D No. of pixels above rethreshold 1  --/U pixels  --/C NUMBER  --/F Areal_1_profile  --/G allDetection::aProf1
aProf2          int not null,       --/D No. of pixels above rethreshold 2  --/U pixels  --/C NUMBER  --/F Areal_2_profile  --/I (see glossary entry for aProf1)
aProf3          int not null,       --/D No. of pixels above rethreshold 3  --/U pixels  --/C NUMBER  --/F Areal_3_profile  --/I (see glossary entry for aProf1)
aProf4          int not null,       --/D No. of pixels above rethreshold 4  --/U pixels  --/C NUMBER  --/F Areal_4_profile  --/I (see glossary entry for aProf1)
aProf5          int not null,       --/D No. of pixels above rethreshold 5  --/U pixels  --/C NUMBER  --/F Areal_5_profile  --/I (see glossary entry for aProf1)
aProf6          int not null,       --/D No. of pixels above rethreshold 6  --/U pixels  --/C NUMBER  --/F Areal_6_profile  --/I (see glossary entry for aProf1)
aProf7          int not null,       --/D No. of pixels above rethreshold 7  --/U pixels  --/C NUMBER  --/F Areal_7_profile  --/I (see glossary entry for aProf1)
aProf8          int not null,       --/D No. of pixels above rethreshold 8  --/U pixels  --/C NUMBER  --/F Areal_8_profile  --/I (see glossary entry for aProf1)
pHeight         real not null,      --/D Highest pixel value above sky  --/U ADU  --/C PHOT_COUNTS_MISC  --/F Peak_height  --/I In counts relative to local value of sky - also zeroth order aperture flux.
pHeightErr      real not null,      --/D Error in peak height  --/U ADU  --/C ERROR  --/F Peak_height_err
aperFlux1       real not null,      --/D Default aperture flux counts 1  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_1  --/I 1/2 &#215; rcore  --/G allDetection::aperFlux1
aperFlux1err    real not null,      --/D Error in aperture flux counts 1  --/U ADU  --/C ERROR  --/F Aper_flux_1_err
aperMag1        real not null,      --/D Calibrated and corrected aperture magnitude 1  --/U mag  --/C PHOT_INT-MAG  --/Q aperFlux1  --/K BINTABLE.APCOR1  --/O
aperMag1err     real not null,      --/D Error in calibrated aperture magnitude 1  --/U mag  --/C ERROR  --/Q aperFlux1,aperFlux1err
aperFlux2       real not null,      --/D Default aperture flux 2  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_2  --/I 1/&#8730;2 &#215; rcore  (see glossary entry for aperFlux1)
aperFlux2err    real not null,      --/D Error in aperture flux 2  --/U ADU  --/C ERROR  --/F Aper_flux_2_err
aperMag2        real not null,      --/D Calibrated and corrected aperture magnitude 2  --/U mag  --/C PHOT_INT-MAG  --/Q aperFlux2  --/K BINTABLE.APCOR2  --/O
aperMag2err     real not null,      --/D Error in calibrated aperture magnitude 2  --/U mag  --/C ERROR  --/Q aperFlux2,aperFlux2err
aperFlux3       real not null,      --/D default aperture flux 3  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_3  --/I rcore; recommended if a single number is required to represent the flux for ALL images.  (see glossary entry for aperFlux1)
aperFlux3err    real not null,      --/D Error in aperture flux 3  --/U ADU  --/C ERROR  --/F Aper_flux_3_err
aperMag3        real not null,      --/D Calibrated and corrected aperture magnitude 3  --/U mag  --/C PHOT_INT-MAG  --/Q aperFlux3  --/K BINTABLE.APCOR3  --/O
aperMag3err     real not null,      --/D Error in calibrated aperture magnitude 3  --/U mag  --/C ERROR  --/Q aperFlux3,aperFlux3err
aperFlux4       real not null,      --/D default aperture flux 4  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_4  --/I &#8730;2 &#215; rcore; ensures &#8764;99% of PSF flux.  (see glossary entry for aperFlux1)
aperFlux4err    real not null,      --/D Error in aperture flux 4  --/U ADU  --/C ERROR  --/F Aper_flux_4_err
aperMag4        real not null,      --/D Calibrated and corrected aperture magnitude 4  --/U mag  --/C PHOT_INT-MAG  --/Q aperFlux4  --/K BINTABLE.APCOR4  --/O
aperMag4err     real not null,      --/D Error in calibrated aperture magnitude 4  --/U mag  --/C ERROR  --/Q aperFlux4,aperFlux4err
aperFlux5       real not null,      --/D default aperture flux 5  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_5  --/I 2 &#215; rcore  (see glossary entry for aperFlux1)
aperFlux5err    real not null,      --/D Error in aperture flux 5  --/U ADU  --/C ERROR  --/F Aper_flux_5_err
aperMag5        real not null,      --/D Calibrated and corrected aperture magnitude 5  --/U mag  --/C PHOT_INT-MAG  --/Q aperFlux5  --/K BINTABLE.APCOR5  --/O
aperMag5err     real not null,      --/D Error in calibrated aperture magnitude 5  --/U mag  --/C ERROR  --/Q aperFlux5,aperFlux5err
aperFlux6       real not null,      --/D default aperture flux 6  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_6  --/I 2&#8730;2 &#215; rcore  (see glossary entry for aperFlux1)
aperFlux6err    real not null,      --/D Error in aperture flux 6  --/U ADU  --/C ERROR  --/F Aper_flux_6_err
aperMag6        real not null,      --/D Calibrated and corrected aperture magnitude 6  --/U mag  --/C PHOT_INT-MAG  --/Q aperFlux6  --/K BINTABLE.APCOR6  --/O
aperMag6err     real not null,      --/D Error in calibrated aperture magnitude 6  --/U mag  --/C ERROR  --/Q aperFlux6,aperFlux6err
aperFlux7       real not null,      --/D default aperture flux 7  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_7  --/I 4 &#215; rcore  (see glossary entry for aperFlux1)
aperFlux7err    real not null,      --/D Error in aperture flux 7  --/U ADU  --/C ERROR  --/F Aper_flux_7_err
```

```
aperMag7        real not null,      --/D Calibrated and corrected aperture magnitude 7  --/U mag  --/C PHOT_INT-MAG  --/Q aperFlux7  --/K BINTABLE.APCOR7  --/O
aperMag7err     real not null,      --/D Error in calibrated aperture magnitude 7  --/U mag  --/C ERROR  --/Q aperFlux7,aperFlux7err
aperFlux8       real not null,      --/D default aperture flux 8  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_8  --/I 5 &#215; rcore  (see glossary entry for aperFlux1)
aperFlux8err    real not null,      --/D Error in aperture flux 8  --/U ADU  --/C ERROR  --/F Aper_flux_8_err
aperMag8        real not null,      --/D Calibrated aperture magnitude 8  --/U mag  --/C PHOT_INT-MAG  --/Q aperFlux8
aperMag8err     real not null,      --/D Error in calibrated aperture magnitude 8  --/U mag  --/C ERROR  --/Q aperFlux8,aperFlux8err
aperFlux9       real not null,      --/D default aperture flux 9  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_9  --/I 6 &#215; rcore  (see glossary entry for aperFlux1)
aperFlux9err    real not null,      --/D Error in aperture flux 9  --/U ADU  --/C ERROR  --/F Aper_flux_9_err
aperMag9        real not null,      --/D Calibrated aperture magnitude 9  --/U mag   --/C PHOT_INT-MAG  --/Q aperFlux9
aperMag9err     real not null,      --/D Error in calibrated aperture magnitude 9  --/U mag   --/C ERROR  --/Q aperFlux9,aperFlux9err
aperFlux10      real not null,      --/D default aperture flux 10  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_10  --/I 7 &#215; rcore  (see glossary entry for aperFlux1)
aperFlux10err   real not null,      --/D Error in aperture flux 10  --/U ADU  --/C ERROR  --/F Aper_flux_10_err
aperMag10       real not null,      --/D Calibrated aperture magnitude 10  --/U mag  --/C PHOT_INT-MAG  --/Q aperFlux10
aperMag10err    real not null,      --/D Error in calibrated aperture magnitude 10  --/U mag  --/C ERROR  --/Q aperFlux10,aperFlux10err
aperFlux11      real not null,      --/D default aperture flux  11  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_11  --/I 8 &#215; rcore  (see glossary entry for aperFlux1)
aperFlux11err   real not null,      --/D Error in aperture flux 11  --/U ADU  --/C ERROR  --/F Aper_flux_11_err
aperMag11       real not null,      --/D Calibrated aperture magnitude 11  --/U mag  --/C PHOT_INT-MAG  --/Q aperFlux11
aperMag11err    real not null,      --/D Error in calibrated aperture magnitude 11  --/U mag  --/C ERROR  --/Q aperFlux11,aperFlux11err
aperFlux12      real not null,      --/D default aperture flux 12  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_12  --/I 10 &#215; rcore  (see glossary entry for aperFlux1)
aperFlux12err   real not null,      --/D Error in aperture flux 12  --/U ADU  --/C ERROR  --/F Aper_flux_12_err
aperMag12       real not null,      --/D Calibrated aperture magnitude 12  --/U mag  --/C PHOT_INT-MAG  --/Q aperFlux12
aperMag12err    real not null,      --/D Error in calibrated aperture magnitude 12  --/U mag  --/C ERROR  --/Q aperFlux12,aperFlux12err
aperFlux13      real not null,      --/D default aperture flux 13  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Aper_flux_13  --/I 12 &#215; rcore ie. &#8776;24 arcsec diameter. (see glossary entry for aperFlux1)
aperFlux13err   real not null,      --/D Error in aperture flux 13  --/U ADU  --/C ERROR  --/F Aper_flux_13_err
aperMag13       real not null,      --/D Calibrated aperture magnitude 13  --/U mag  --/C PHOT_INT-MAG  --/Q aperFlux13
aperMag13err    real not null,      --/D Error in calibrated aperture magnitude 13  --/U mag  --/C ERROR  --/Q aperFlux13,aperFlux13err
petroRad        real not null,      --/D r_p as defined in Yasuda et al. 2001 AJ 112 1104  --/U pixels  --/C EXTENSION_RAD  --/F Petr_radius
kronRad         real not null,      --/D r_k as defined in Bertin and Arnouts 1996 A&#38;A Supp 117 393  --/U pixels  --/C EXTENSION_RAD  --/F Kron_radius
hallRad         real not null,      --/D r_h image scale radius eg. Hall &#38; Mackay 1984 MNRAS 210 979  --/U pixels  --/C EXTENSION_RAD  --/F Hall_radius
petroFlux       real not null,      --/D flux within circular aperture to k &#215; r_p ; k = 2  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Petr_flux
petroFluxErr    real not null,      --/D error on Petrosian flux  --/U ADU  --/C ERROR  --/F Petr_flux_err
petroMag        real not null,      --/D Calibrated Petrosian magnitude within circular aperture r_p  --/U mag  --/C PHOT_INT-MAG  --/Q petroFlux
petroMagErr     real not null,      --/D error on calibrated Petrosian magnitude  --/U mag  --/C ERROR  --/Q petroFlux,petroFluxErr
kronFlux        real not null,      --/D flux within circular aperture to k &#215; r_k ; k = 2  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Kron_flux
kronFluxErr     real not null,      --/D error on Kron flux --/U ADU   --/C ERROR   --/F Kron_flux_err
kronMag         real not null,      --/D Calibrated Kron magnitude within circular aperture r_k  --/U mag  --/C PHOT_INT-MAG  --/Q kronFlux
kronMagErr      real not null,      --/D error on calibrated Kron magnitude  --/U mag  --/C ERROR  --/Q kronFlux,kronFluxErr
hallFlux        real not null,      --/D flux within circular aperture to k &#215; r_h; k = 5; alternative total flux  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F Hall_flux
hallFluxErr     real not null,      --/D error on Hall flux  --/U ADU  --/C ERROR  --/F Hall_flux_err
hallMag         real not null,      --/D Calibrated magnitude within circular aperture r_hall  --/U mag  --/C PHOT_INT-MAG  --/Q hallFlux
hallMagErr      real not null,      --/D Calibrated error on Hall magnitude  --/U mag  --/C ERROR  --/Q hallFlux,hallFluxErr
errBits         int not null,       --/D processing warning/error bitwise flags  --/C CODE_MISC  --/F Error_bit_flag
sky             real not null,      --/D local interpolated sky level from background tracker  --/U ADU  --/C INST_SKY_LEVEL  --/F Sky_level
skyVar          real not null,      --/D local estimate of variation in sky level around image  --/U ADU  --/C INST_SKY_LEVEL  --/F Sky_rms
deblend         int not null,       --/D flag for parent or part of deblended deconstruct (redundant since only deblended images are kept)  --/C CODE_MISC  --/F Parent_or_child
ra              float not null,     --/D Celestial Right Ascension  --/U Degrees  --/C POS_EQ_RA_MAIN  --/Q extNum,x,y  --/G allDetection::ra
dec             float not null,     --/D Celestial Declination  --/U Degrees  --/C POS_EQ_DEC_MAIN  --/Q extNum,x,y  --/G allDetection::dec
cx              float not null,     --/D unit vector of spherical co-ordinates  --/C POS_EQ_X  --/Q ra,dec
cy              float not null,     --/D unit vector of spherical co-ordinates  --/C POS_EQ_Y  --/Q ra,dec
cz              float not null,     --/D unit vector of spherical co-ordinates  --/C POS_EQ_Z  --/Q ra,dec
htmID           bigint not null,    --/D Hierarchical Triangular Mesh (HTM) index, 20 deep, for equatorial co-ordinates  --/C POS_GENERAL  --/L 20  --/Q ra,dec
```

```
l               float not null,     --/D Galactic longitude --/U Degrees --/C POS_GAL_LON  --/Q ra,dec
b               float not null,     --/D Galactic latitude  --/U Degrees --/C POS_GAL_LAT  --/Q ra,dec
lambda          float not null,     --/D SDSS system spherical co-ordinate 1  --/U Degrees  --/C POS  --/Q ra,dec
eta             float not null,     --/D SDSS system spherical co-ordinate 2  --/U Degrees  --/C POS  --/Q ra,dec
class           smallint not null,  --/D Flag indicating most probable morphological classification  --/C CLASS_MISC  --/F Classification  --/I -1 stellar, +1 non--stellar, 0 noise, -2 borderline stellar, -3 boderline galaxy, -9 saturated  --/G allDetection::class
classStat       real not null,      --/D N(0,1) stellarness-of-profile statistic  --/C STAT_PROP  --/F Statistic  --/G allDetection::classStat
psfFlux         real not null,      --/D PSF-fitted flux  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F PSF_flux  --/N -0.9999995e9
psfFluxErr      real not null,      --/D Error on PSF-fitted flux  --/U ADU  --/C ERROR  --/F PSF_flux_err  --/N -0.9999995e9
psfMag          real not null,      --/D PSF-fitted calibrated magnitude  --/U mag  --/C PHOT_PROFILE  --/N -0.9999995e9  --/Q psfFlux
psfMagErr       real not null,      --/D Error on PSF-fitted calibrated magnitude  --/U mag  --/C ERROR  --/N -0.9999995e9  --/Q psfFlux,psfFluxErr
psfFitX         real not null,      --/D PSF-fitted X coordinate  --/U pixels  --/C POS_PLATE_X  --/F PSF_fit_X  --/N -0.9999995e9
psfFitXerr      real not null,      --/D Error on PSF-fitted X coordinate  --/U pixels  --/C ERROR  --/F PSF_fit_X_err  --/N -0.9999995e9
psfFitY         real not null,      --/D PSF-fitted Y coordinate  --/U pixels  --/C POS_PLATE_Y  --/F PSF_fit_Y  --/N -0.9999995e9
psfFitYerr      real not null,      --/D Error on PSF-fitted Y coordinate  --/U pixels  --/C ERROR  --/F PSF_fit_y_err  --/N -0.9999995e9
psfFitChi2      real not null,      --/D standard normalised variance of PSF fit  --/C FIT_STDEV  --/F PSF_fit_chi2  --/N -0.9999995e9
psfFitDof       smallint not null,  --/D no. of degrees of freedom of PSF fit  --/C STAT_N-DOF  --/F PSF_fit_dof  --/N -9999
SerFlux1D       real not null,      --/D  1D Sersic flux  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F 1D_Sersic_flux  --/N -0.9999995e9  --/I [For numerical stability the Sersic fits will use the previously derived x-y coordinates]
SerMag1D        real not null,      --/D  Calibrated 1D Sersic flux  --/U mag  --/C PHOT_PROFILE  --/N -0.9999995e9  --/Q SerFlux1D  --/I [For numerical stability the Sersic fits will use the previously derived x-y coordinates]
SerScaleLen1D   real not null,      --/D  Scale length  --/C PHOT_PROFILE_PAR  --/F 1D_Sersic_scale_len  --/N -0.9999995e9  --/I [For numerical stability the Sersic fits will use the previously derived x-y coordinates]
SerIdx1D        real not null,      --/D  Power law index  --/C PHOT_PROFILE_PAR  --/F 1D_Sersic_index  --/N -0.9999995e9  --/I [For numerical stability the Sersic fits will use the previously derived x-y coordinates]
SerFit1DChi2    real not null,      --/D  Error in 1D fit  --/C PHOT_PROFILE_PAR  --/F 1D_Sersic_fit_chi2  --/N -0.9999995e9  --/I standard normalised variance of fit [For numerical stability the Sersic fits will use the previously derived x-y coordinates]
SerFitNu1D      real not null,      --/D No. of degrees of freedom for 1D Sersic fit  --/C STAT_N-DOF  --/F 1D_Sersic_fit_nu  --/N -0.9999995e9  --/I [For numerical stability the Sersic fits will use the previously derived x-y coordinates]
SerFlux2D       real not null,      --/D  2D Sersic flux  --/U ADU  --/C PHOT_INTENSITY_ADU  --/F 2D_Sersic_flux  --/N -0.9999995e9  --/I [For numerical stability the Sersic fits will use the previously derived x-y coordinates]
SerMag2D        real not null,      --/D  Calibrated 2D Sersic flux  --/U mag  --/C PHOT_PROFILE  --/N -0.9999995e9  --/Q SerFlux2D  --/I [For numerical stability the Sersic fits will use the previously derived x-y coordinates]
SerScaleLen2D   real not null,      --/D  Scale length  --/C PHOT_PROFILE_PAR  --/F 2D_Sersic_scale_len  --/N -0.9999995e9  --/I [For numerical stability the Sersic fits will use the previously derived x-y coordinates]
SerIdx2D        real not null,      --/D  Power law index  --/C PHOT_PROFILE_PAR  --/F 2D_Sersic_index  --/N -0.9999995e9  --/I [For numerical stability the Sersic fits will use the previously derived x-y coordinates]
SerFit2DChi2    real not null,      --/D  Error in 2D fit  --/C PHOT_PROFILE_PAR  --/F 2D_Sersic_fit_chi2  --/N -0.9999995e9  --/I standard normalised variance of fit [For numerical stability the Sersic fits will use the previously derived x-y coordinates]
SerFitNu2D      real not null,      --/D No. of degrees of freedom for 2D Sersic fit  --/C STAT_N-DOF  --/F 2D_Sersic_fit_nu  --/N -0.9999995e9  --/I [For numerical stability the Sersic fits will use the previously derived x-y coordinates]
--
--  Additional   (non-CASU) attributes:
--
ppErrBits     int not null,         --/D additional WFAU post-processing error bits (place holder for now)  --/C CODE_MISC  --/N 0
deprecated    tinyint not null,     --/D Code for a current (=0) or deprecated (!=0) detection  --/C CODE_MISC  --/N 0  --/G allTables::deprecated
objID bigint not null,              --/D Unique identifier for this detection  --/C ID_MAIN
CONSTRAINT uid_vhsDetection UNIQUE (objID),
CONSTRAINT pk_vhsDetection PRIMARY KEY (multiframeID,extNum,seqNum)
) ON Detection_FG
GO
CREATE TABLE vhsSource(
-------------------------------------------------------------------------------
--/H Contains VHS merged sources from VHS detections in vhsDetection.
--
--/T Prescription for the merging (from RequiredFilters) is: Y,Jx2,H,K.
--
--/T Required constraints: primary key is (sourceID)
--/T                       (frameSetID) references vhsMergeLog(frameSetID)
-------------------------------------------------------------------------------
sourceID        bigint not null,    --/D UID (unique over entire WSA via programme ID prefix) of this merged detection as assigned by merge algorithm  --/C ID_MAIN
cuEventID       int not null,       --/D UID of curation event giving rise to this record  --/C REFER_CODE
frameSetID      bigint not null,    --/D UID of the set of frames that this merged source comes from  --/C REFER_CODE  --/G allSource::frameSetID
ra              float not null,     --/D Celestial Right Ascension  --/U Degrees  --/C POS_EQ_RA_MAIN  --/G allDetection::ra
```

```
dec             float not null,     --/D Celestial Declination  --/U Degrees  --/C POS_EQ_DEC_MAIN  --/G allDetection::dec
sigRa           real not null,      --/D Uncertainty in RA  --/U Degrees  --/C ERROR  --/N -0.9999995e9
sigDec          real not null,      --/D Uncertainty in Dec  --/U Degrees  --/C ERROR  --/N -0.9999995e9
epoch           float not null,      --/D Epoch of position measurement  --/U Years  --/C TIME_EPOCH   --/N -0.9999995e9
muRa            real not null,      --/D Proper motion in RA direction  --/U mas/yr  --/N -0.9999995e9  --/C POS_EQ_PMRA
muDec           real not null,      --/D Proper motion in Dec direction  --/U mas/yr  --/N -0.9999995e9  --/C POS_EQ_PMDEC
sigMuRa         real not null,      --/D Error on proper motion in RA direction  --/U mas/yr  --/N -0.9999995e9  --/C ERROR
sigMuDec        real not null,      --/D Error on proper motion in Dec direction  --/U mas/yr  --/N -0.9999995e9  --/C ERROR
chi2            real not null,      --/D Chi-squared value of proper motion solution  --/N -0.9999995e9  --/C FIT_CHI2
nFrames         tinyint not null,   --/D No. of frames used for this proper motion measurement  --/C NUMBER  --/N 0
cx              float not null,     --/D unit vector of spherical co-ordinates  --/C POS_EQ_X  --/Q ra,dec
cy              float not null,     --/D unit vector of spherical co-ordinates  --/C POS_EQ_Y  --/Q ra,dec
cz              float not null,     --/D unit vector of spherical co-ordinates  --/C POS_EQ_Z  --/Q ra,dec
htmID           bigint not null,    --/D Hierarchical Triangular Mesh (HTM) index, 20 deep, for equatorial co-ordinates  --/C POS_GENERAL  --/L 20  --/Q ra,dec
l               float not null,     --/D Galactic longitude  --/U Degrees  --/C POS_GAL_LON  --/Q ra,dec
b               float not null,     --/D Galactic latitude  --/U Degrees  --/C POS_GAL_LAT  --/Q ra,dec
lambda          float not null,     --/D SDSS system spherical co-ordinate 1  --/U Degrees  --/C POS  --/Q ra,dec
eta             float not null,     --/D SDSS system spherical co-ordinate 2  --/U Degrees  --/C POS  --/Q ra,dec
priOrSec        bigint not null,    --/D Seam code for a unique (=0) or duplicated (!=0) source (eg. flags overlap duplicates).  --/C CODE_MISC  --/N -99999999  --/G allSource::priOrSec
zmyPnt      real not null,      --/D Point source colour Z-Y (using aperMag3)  --/U mag  --/C PHOT_COLOR  --/Q zAperMag3,yAperMag3  --/N -0.9999995e9  --/G allSource::colours
zmyPntErr   real not null,      --/D Error on point source colour Z-Y  --/U mag  --/C ERROR  --/Q zAperMag3Err,yAperMag3Err  --/N -0.9999995e9  --/G allSource::colours
ymjPnt      real not null,      --/D Point source colour Y-J (using aperMag3)  --/U mag  --/C PHOT_COLOR  --/Q yAperMag3,jAperMag3  --/N -0.9999995e9  --/G allSource::colours
ymjPntErr   real not null,      --/D Error on point source colour Y-J  --/U mag  --/C ERROR  --/Q yAperMag3Err,jAperMag3Err  --/N -0.9999995e9  --/G allSource::colours
jmkPnt      real not null,      --/D Point source colour J-K (using aperMag3)  --/U mag  --/C PHOT_COLOR  --/Q jAperMag3,kAperMag3  --/N -0.9999995e9  --/G allSource::colours
jmkPntErr   real not null,      --/D Error on point source colour J-K  --/U mag  --/C ERROR  --/Q jAperMag3Err,kAperMag3Err  --/N -0.9999995e9  --/G allSource::colours
zmyExt      real not null,      --/D Extended source colour Z-Y (using aperMag3)  --/U mag  --/C PHOT_COLOR  --/Q zAperMag3,yAperMag3  --/N -0.9999995e9 --/G allSource::colours
zmyExtErr   real not null,      --/D Error on extended source colour Z-Y  --/U mag  --/C ERROR  --/Q zAperMag3Err,yAperMag3Err  --/N -0.9999995e9  --/G allSource::colours
ymjExt      real not null,      --/D Extended source colour Y-J (using aperMag3)  --/U mag  --/C PHOT_COLOR  --/Q yAperMag3,jAperMag3  --/N -0.9999995e9 --/G allSource::colours
ymjExtErr   real not null,      --/D Error on extended source colour Y-J  --/U mag  --/C ERROR  --/Q yAperMag3Err,jAperMag3Err  --/N -0.9999995e9  --/G allSource::colours
jmkExt      real not null,      --/D Extended source colour J-K (using aperMag3)  --/U mag  --/C PHOT_COLOR  --/Q jAperMag3,kAperMag3  --/N -0.9999995e9 --/G allSource::colours
jmkExtErr   real not null,      --/D Error on extended source colour J-K  --/U mag  --/C ERROR  --/Q jAperMag3Err,kAperMag3Err  --/N -0.9999995e9  --/G allSource::colours
mergedClassStat real not null,      --/D Merged N(0,1) stellarness-of-profile statistic  --/C STAT_PROP  --/N -0.9999995e9  --/G allSource::mergedClassStat
mergedClass     smallint not null,  --/D Class flag from available measurements (1|0|-1|-2|-3|-9=galaxy|noise|stellar|probableStar|probableGalaxy|saturated)  --/C CODE_MISC --/G allSource::mergedClass
pStar           real not null,      --/D Probability that the source is a star  --/C STAT_PROP  --/G allSource::mergedClass
pGalaxy         real not null,      --/D Probability that the source is a galaxy  --/C STAT_PROP  --/G allSource::mergedClass
pNoise          real not null,      --/D Probability that the source is noise  --/C STAT_PROP  --/G allSource::mergedClass
pSaturated      real not null,      --/D Probability that the source is saturated  --/C STAT_PROP  --/G allSource::mergedClass
eBV             real not null, --/D The galactic dust extinction value measured from the Schlegel, Finkbeiner & Davis (1998) maps. --/N -0.9999995e9
aZ           real not null, --/D The galactic extinction correction in the Z band for extragalactic objects  --/U mag  --/N -0.9999995e9
aY           real not null, --/D The galactic extinction correction in the J band for extragalactic objects  --/U mag  --/N -0.9999995e9
aJ           real not null, --/D The galactic extinction correction in the H band for extragalactic objects  --/U mag  --/N -0.9999995e9
aK           real not null, --/D The galactic extinction correction in the K band for extragalactic objects  --/U mag  --/N -0.9999995e9
--
--  Z parameters from detection table:
--
zHallMag      real not null,      --/D Total point source Z mag  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
zHallMagErr   real not null,      --/D Error in total point source Z mag  --/U mag  --/C ERROR  --/N -0.9999995e9
zPetroMag     real not null,      --/D Extended source Z mag (Petrosian)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
zPetroMagErr  real not null,      --/D Error in extended source Z mag (Petrosian)  --/U mag  --/C ERROR  --/N -0.9999995e9
zPsfMag       real not null,      --/D Point source profile-fitted Z mag  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
zPsfMagErr    real not null,      --/D Error in point source profile-fitted Z mag  --/U mag  --/C ERROR  --/N -0.9999995e9
```

```
zSerMag2D     real not null,      --/D Extended source Z mag (profile-fitted)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
zSerMag2DErr  real not null,      --/D Error in extended source Z mag (profile-fitted)  --/U mag  --/C ERROR  --/N -0.9999995e9
zAperMag3     real not null,      --/D Default point/extended source Z aperture corrected mag  (2.0 arcsec aperture diameter)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9  --/I If in doubt use this flux estimator
zAperMag3Err  real not null,      --/D Error in default point/extended source Z mag (2.0 arcsec aperture diameter)  --/U mag  --/C ERROR  --/N -0.9999995e9
zAperMag4     real not null,      --/D Extended source Z aperture corrected mag  (2.8 arcsec aperture diameter)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
zAperMag4Err  real not null,      --/D Error in extended source Z mag (2.8 arcsec aperture diameter)  --/U mag  --/C ERROR  --/N -0.9999995e9
zAperMag6     real not null,      --/D Extended source Z aperture corrected mag  (5.7 arcsec aperture diameter)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
zAperMag6Err  real not null,      --/D Error in extended source Z mag (5.7 arcsec aperture diameter)  --/U mag  --/C ERROR  --/N -0.9999995e9
zGausig       real not null,      --/D RMS of axes of ellipse fit in Z  --/U pixels  --/C MORPH_PARAM  --/N -0.9999995e9
zEll          real not null,      --/D 1-b/a, where a/b=semi-major/minor axes in Z  --/C PHYS_ELLIPTICITY --/N -0.9999995e9
zPA           real not null,      --/D ellipse fit celestial orientation in Z  --/U Degrees  --/C POS_POS-ANG  --/N -0.9999995e9
zErrBits      int not null,       --/D processing warning/error bitwise flags in Z  --/C CODE_MISC  --/N -99999999
zDeblend      int not null,       --/D flag indicating parent/child relation in Z  --/C CODE_MISC  --/N -99999999
zClass        smallint not null,  --/D discrete image classification flag in Z  --/C CLASS_MISC  --/N -9999
zClassStat    real not null,      --/D N(0,1) stellarness-of-profile statistic in Z  --/C STAT_PROP  --/N -0.9999995e9
zppErrBits    int not null,       --/D additional WFAU post-processing error bits in Z  --/C CODE_MISC  --/N 0
zSeqNum       int not null,       --/D the running number of the Z detection  --/C ID_NUMBER  --/N -99999999
zObjID        bigint not null,    --/D the UID of the Z detection  --/C ID_NUMBER  --/N -99999999
zXi           real not null,      --/D Offset of Z detection from master position (+east/-west)  --/C POS_EQ_RA_OFF  --/N -0.9999995e9  --/U arcsec  --/G allSource::XiEta
zEta          real not null,      --/D Offset of Z detection from master position (+north/-south)  --/C POS_EQ_DEC_OFF  --/N -0.9999995e9  --/U arcsec  --/G allSource::XiEta
--
--  Y parameters from detection table:
--
yHallMag      real not null,      --/D Total point source 1st epoch Y mag  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
yHallMagErr   real not null,      --/D Error in total point source 1st epoch Y mag  --/U mag  --/C ERROR  --/N -0.9999995e9
yPetroMag     real not null,      --/D Extended source 1st epoch Y mag (Petrosian)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
yPetroMagErr  real not null,      --/D Error in extended source 1st epoch Y mag (Petrosian)  --/U mag  --/C ERROR  --/N -0.9999995e9
yPsfMag       real not null,      --/D Point source profile-fitted 1st epoch Y mag  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
yPsfMagErr    real not null,      --/D Error in point source profile-fitted 1st epoch Y mag  --/U mag  --/C ERROR  --/N -0.9999995e9
ySerMag2D     real not null,      --/D Extended source 1st epoch Y mag (profile-fitted)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
ySerMag2DErr  real not null,      --/D Error in extended source 1st epoch Y mag (profile-fitted)  --/U mag  --/C ERROR  --/N -0.9999995e9
yAperMag3     real not null,      --/D Default point/extended source 1st epoch Y aperture corrected mag  (2.0 arcsec aperture diameter)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9  --/I If in doubt use this flux estimator
yAperMag3Err  real not null,      --/D Error in default point/extended source 1st epoch Y mag (2.0 arcsec aperture diameter)  --/U mag  --/C ERROR  --/N -0.9999995e9
yAperMag4     real not null,      --/D Extended source 1st epoch Y aperture corrected mag  (2.8 arcsec aperture diameter)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
yAperMag4Err  real not null,      --/D Error in extended source 1st epoch Y mag (2.8 arcsec aperture diameter)  --/U mag  --/C ERROR  --/N -0.9999995e9
yAperMag6     real not null,      --/D Extended source 1st epoch Y aperture corrected mag  (5.7 arcsec aperture diameter)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
yAperMag6Err  real not null,      --/D Error in extended source 1st epoch Y mag (5.7 arcsec aperture diameter)  --/U mag  --/C ERROR  --/N -0.9999995e9
yGausig       real not null,      --/D RMS of axes of ellipse fit in 1st epoch Y  --/U pixels  --/C MORPH_PARAM  --/N -0.9999995e9
yEll          real not null,      --/D 1-b/a, where a/b=semi-major/minor axes in 1st epoch Y  --/C PHYS_ELLIPTICITY --/N -0.9999995e9
yPA           real not null,      --/D ellipse fit celestial orientation in 1st epoch Y  --/U Degrees  --/C POS_POS-ANG  --/N -0.9999995e9
yErrBits      int not null,       --/D processing warning/error bitwise flags in 1st epoch Y  --/C CODE_MISC  --/N -99999999
yDeblend      int not null,       --/D flag indicating parent/child relation in 1st epoch Y  --/C CODE_MISC  --/N -99999999
yClass        smallint not null,  --/D discrete image classification flag in 1st epoch Y  --/C CLASS_MISC  --/N -9999
yClassStat    real not null,      --/D N(0,1) stellarness-of-profile statistic in 1st epoch Y  --/C STAT_PROP  --/N -0.9999995e9
yppErrBits    int not null,       --/D additional WFAU post-processing error bits in 1st epoch Y  --/C CODE_MISC  --/N 0
ySeqNum       int not null,       --/D the running number of the 1st epoch Y detection  --/C ID_NUMBER  --/N -99999999
yObjID        bigint not null,    --/D the UID of the 1st epoch Y detection  --/C ID_NUMBER  --/N -99999999
yXi           real not null,      --/D Offset of 1st epoch Y detection from master position (+east/-west)  --/C POS_EQ_RA_OFF  --/N -0.9999995e9  --/U arcsec  --/G allSource::XiEta
yEta          real not null,      --/D Offset of 1st epoch Y detection from master position (+north/-south)  --/C POS_EQ_DEC_OFF  --/N -0.9999995e9  --/U arcsec  --/G allSource::XiEta
--
--  J parameters from detection table:
```

```
--
jHallMag      real not null,      --/D Total point source J mag  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
jHallMagErr   real not null,      --/D Error in total point source J mag  --/U mag  --/C ERROR  --/N -0.9999995e9
jPetroMag     real not null,      --/D Extended source J mag (Petrosian)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
jPetroMagErr  real not null,      --/D Error in extended source J mag (Petrosian)  --/U mag  --/C ERROR  --/N -0.9999995e9
jPsfMag       real not null,      --/D Point source profile-fitted J mag  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
jPsfMagErr    real not null,      --/D Error in point source profile-fitted J mag  --/U mag  --/C ERROR  --/N -0.9999995e9
jSerMag2D     real not null,      --/D Extended source J mag (profile-fitted)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
jSerMag2DErr  real not null,      --/D Error in extended source J mag (profile-fitted)  --/U mag  --/C ERROR  --/N -0.9999995e9
jAperMag3     real not null,      --/D Default point/extended source J aperture corrected mag  (2.0 arcsec aperture diameter)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9  --/I If in doubt use this flux estimator
jAperMag3Err  real not null,      --/D Error in default point/extended source J mag (2.0 arcsec aperture diameter)  --/U mag  --/C ERROR  --/N -0.9999995e9
jAperMag4     real not null,      --/D Extended source J aperture corrected mag  (2.8 arcsec aperture diameter)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
jAperMag4Err  real not null,      --/D Error in extended source J mag (2.8 arcsec aperture diameter)  --/U mag  --/C ERROR  --/N -0.9999995e9
jAperMag6     real not null,      --/D Extended source J aperture corrected mag  (5.7 arcsec aperture diameter)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
jAperMag6Err  real not null,      --/D Error in extended source J mag (5.7 arcsec aperture diameter)  --/U mag  --/C ERROR  --/N -0.9999995e9
jGausig       real not null,      --/D RMS of axes of ellipse fit in J  --/U pixels  --/C MORPH_PARAM  --/N -0.9999995e9
jEll          real not null,      --/D 1-b/a, where a/b=semi-major/minor axes in J  --/C PHYS_ELLIPTICITY --/N -0.9999995e9
jPA           real not null,      --/D ellipse fit celestial orientation in J  --/U Degrees  --/C POS_POS-ANG  --/N -0.9999995e9
jErrBits      int not null,       --/D processing warning/error bitwise flags in J  --/C CODE_MISC  --/N -99999999
jDeblend      int not null,       --/D flag indicating parent/child relation in J  --/C CODE_MISC  --/N -99999999
jClass        smallint not null,  --/D discrete image classification flag in J  --/C CLASS_MISC  --/N -9999
jClassStat    real not null,      --/D N(0,1) stellarness-of-profile statistic in J  --/C STAT_PROP  --/N -0.9999995e9
jppErrBits    int not null,       --/D additional WFAU post-processing error bits in J  --/C CODE_MISC  --/N 0
jSeqNum       int not null,       --/D the running number of the J detection  --/C ID_NUMBER  --/N -99999999
jObjID        bigint not null,    --/D the UID of the J detection  --/C ID_NUMBER  --/N -99999999
jXi           real not null,      --/D Offset of J detection from master position (+east/-west)  --/C POS_EQ_RA_OFF  --/N -0.9999995e9  --/U arcsec  --/G allSource::XiEta
jEta          real not null,      --/D Offset of J detection from master position (+north/-south)  --/C POS_EQ_DEC_OFF  --/N -0.9999995e9  --/U arcsec  --/G allSource::XiEta
--
--  K parameters from detection table:
--
kHallMag      real not null,      --/D Total point source K mag  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
kHallMagErr   real not null,      --/D Error in total point source K mag  --/U mag  --/C ERROR  --/N -0.9999995e9
kPetroMag     real not null,      --/D Extended source K mag (Petrosian)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
kPetroMagErr  real not null,      --/D Error in extended source K mag (Petrosian)  --/U mag  --/C ERROR  --/N -0.9999995e9
kPsfMag       real not null,      --/D Point source profile-fitted K mag  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
kPsfMagErr    real not null,      --/D Error in point source profile-fitted K mag  --/U mag  --/C ERROR  --/N -0.9999995e9
kSerMag2D     real not null,      --/D Extended source K mag (profile-fitted)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
kSerMag2DErr  real not null,      --/D Error in extended source K mag (profile-fitted)  --/U mag  --/C ERROR  --/N -0.9999995e9
kAperMag3     real not null,      --/D Default point/extended source K aperture corrected mag  (2.0 arcsec aperture diameter)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9  --/I If in doubt use this flux estimator
kAperMag3Err  real not null,      --/D Error in default point/extended source K mag (2.0 arcsec aperture diameter)  --/U mag  --/C ERROR  --/N -0.9999995e9
kAperMag4     real not null,      --/D Extended source K aperture corrected mag  (2.8 arcsec aperture diameter)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
kAperMag4Err  real not null,      --/D Error in extended source K mag (2.8 arcsec aperture diameter)  --/U mag  --/C ERROR  --/N -0.9999995e9
kAperMag6     real not null,      --/D Extended source K aperture corrected mag  (5.7 arcsec aperture diameter)  --/U mag  --/C PHOT_MAG  --/N -0.9999995e9
kAperMag6Err  real not null,      --/D Error in extended source K mag (5.7 arcsec aperture diameter)  --/U mag  --/C ERROR  --/N -0.9999995e9
kGausig       real not null,      --/D RMS of axes of ellipse fit in K  --/U pixels  --/C MORPH_PARAM  --/N -0.9999995e9
kEll          real not null,      --/D 1-b/a, where a/b=semi-major/minor axes in K  --/C PHYS_ELLIPTICITY --/N -0.9999995e9
kPA           real not null,      --/D ellipse fit celestial orientation in K  --/U Degrees  --/C POS_POS-ANG  --/N -0.9999995e9
kErrBits      int not null,       --/D processing warning/error bitwise flags in K  --/C CODE_MISC  --/N -99999999
kDeblend      int not null,       --/D flag indicating parent/child relation in K  --/C CODE_MISC  --/N -99999999
kClass        smallint not null,  --/D discrete image classification flag in K  --/C CLASS_MISC  --/N -9999
kClassStat    real not null,      --/D N(0,1) stellarness-of-profile statistic in K  --/C STAT_PROP  --/N -0.9999995e9
```

```
kppErrBits    int not null,       --/D additional WFAU post-processing error bits in K  --/C CODE_MISC  --/N 0
kSeqNum       int not null,       --/D the running number of the K detection  --/C ID_NUMBER  --/N -99999999
kObjID        bigint not null,    --/D the UID of the K detection  --/C ID_NUMBER  --/N -99999999
kXi           real not null,      --/D Offset of K detection from master position (+east/-west)  --/C POS_EQ_RA_OFF  --/N -0.9999995e9  --/U arcsec  --/G allSource::XiEta
kEta          real not null,      --/D Offset of K detection from master position (+north/-south)  --/C POS_EQ_DEC_OFF  --/N -0.9999995e9  --/U arcsec  --/G allSource::XiEta
--
CONSTRAINT pk_vhsSource PRIMARY KEY (sourceID)
) ON Source_FG
GO
```

More scripts are available online from the WSA website [29].

9.4 Constraints

```
--
-- VSAConstraints.sql
--
-- File to add primary and foreign key constraints to the tables of the WSA.
--
-- Original author: Nigel Hambly, WFAU, IfA, University of Edinburgh
--
-- Revision History (CVS repository):
-- Login name of the user who checked in this revision:  $Author: etws $
-- Date and time (UTC) when the revision was checked in: $Date: 2006/09/29 21:33:34 $
-- Login name of the user who locked the revision:       $Locker:  $
-- CVS revision number:                                  $Revision: 1.17 $
--
--
-- vhsDetection foreign keys:
--
IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='fk_vhsDetection_multiframeID_extNum_to_MultiframeDetector_multiframeID_extNum') ALTER TABLE vhsDetection ADD CONSTRAINT fk_vhsDetection_multiframeID_extNum_to_MultiframeDetector_multiframeID_extNum
FOREIGN KEY (multiframeID,extNum) REFERENCES MultiframeDetector(multiframeID,extNum) ON UPDATE NO ACTION ON DELETE CASCADE
--
-- vhsSource foreign keys:
--
IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='fk_vhsSource_frameSetID_to_vhsMergeLog_frameSetID') ALTER TABLE vhsSource ADD CONSTRAINT fk_vhsSource_frameSetID_to_vhsMergeLog_frameSetID
FOREIGN KEY (frameSetID) REFERENCES vhsMergeLog(frameSetID) ON UPDATE NO ACTION ON DELETE CASCADE
--
```

10 ACRONYMS & ABBREVIATIONS

ADnn : Applicable Document No. nn
API : Application Program Interface
BLOB : Binary Large OBject
CASU : Cambridge Astronomical Survey Unit
Combiframe: WSA parlance for any image that is a combination of two or more Multiframes
CSV : Comma Separated Value
CVS : Concurrent Versions System

DBI : Data Base Interface
DBMS : DataBase Management System
DXS : Deep eXtragalactic Survey (UKIDSS)
ELVIS : Emission-Line galaxies with VISTA Survey (VPSP)
ERM : Entity–Relationship Model
ESO : European Southern Observatory
FIRST : Faint Images of the Radio Sky at Twenty-cm (VLA radio survey)
FITS : Flexible Image Transport System
GCS : Galactic Clusters Survey (UKIDSS)
GPS : Galactic Plane Survey (UKIDSS)
HDU : Header Data Unit (FITS nomenclature)
HEALPix : Hierarchical Equal Area isoLatitude PIXelisation
HTM : Hierarchical Triangular Mesh
IAU : International Astronomy Union
ICRS : International Celestial Reference System
IRAS : Infrared Astronomical Satellite (JPL Far-IR satellite)
JAC : Joint Astronomy Centre
LAS : Large Area Survey (UKIDSS)
MEF : Multi–Extension FITS
MGC : Millennium Galaxy Catalogue (Survey on Issac Newton Telescope with supplemental spectroscopic data from many sources
Multiframe : WSA parlance for any frame consisting of several distinct device images
RDBMS : Relational DBMS
ROSAT : ROentgen SATellite (ESO X-ray satellite)
SDSS : Sloan Digitial Sky Survey
SQL : Structured Query Language
UID : Unique IDentifier
UDS : Ultra Deep Survey (UKIDSS)
UKIDSS : UKIRT Deep Infrared Sky Survey
UKIRT : United Kingdom Infrared Telescope
VISTA : Visible and Infrared Survey Telescope for Astronomy
VGPS : VISTA Galactic Plane Survey (VPSP)
VHS : VISTA Hemisphere Survey (VPSP)
VIDEO : VISTA Deep Extragalactic Observations survey (VPSP)
VIKING : VISTA Kilo-Degree Infrared Galaxy survey (VPSP)
VMC : VISTA near-infrared survey of the Magellanic System (VPSP)
VPSP : VISTA Public Survey Proposals
VUUDS : VISTA-UKIRT Ultra Deep Survey (VPSP)
VVV : VISTA Variables in the Via Lactea (VPSP)
WCS : World Co-ordinate System
WFAU : Wide Field Astronomy Unit (Edinburgh)
WSA : WFCAM Science Archive
XMM : Xray Multi-Mirror Telescope (XMM-Newton, ESO X-ray satellite)
2MASS : 2 Micron All–Sky Survey

11 APPLICABLE DOCUMENTS

AD01	VSA Science Requirements Analysis Document [22]	VDF-WFA-VSA-002 Issue: 1.0, 30/09/06
AD02	UKIDSS Proposal [28]	
AD03	VSA Interface Control Document [23]	VDF-WFA-VSA-004 Issue: 1.0, 30/09/06
AD04	VSA Data Flow Document [24]	VDF-WFA-VSA-005 Issue: 1.0, 30/09/06
AD05	VSA Hardware Design Document [25]	VDF-WFA-VSA-006 Issue: 1.0, 30/09/06
AD06	Usages of the WFCAM Science Archive [30]	
AD07	VSA Management and Planning document [27]	VDF-WFA-VSA-003 Issue: 1.0, 30/09/06
AD08	VSA Software Architecture Design document [26]	VDF-WFA-VSA-009 Issue: 1.0, 30/09/06

12 CHANGE RECORD

Issue	**Date**	**Section(s) Affected**	**Description of Change/Change Request Reference/Remarks**
Draft	01/09/06	All	New document
1.0	30/09/06	All	Final issue (for VDFS review)

13 NOTIFICATION LIST

The following people should be notified by email whenever a new version of this document has been issued:

WFAU:	P Williams, N Hambly
CASU:	M Irwin, J Lewis
QMUL:	J Emerson
ATC:	M Stewart
ESO:	P. Quinn