KLOE Bookkeeping Database

I.Sfiligoi
INFN LNF

KLOE memo 226

rev 1.4

Abstract: KLOE is acquiring huge amounts of data. To reduce the complexity of managing all these data, they get stored as files and moved to tape for permanent storage. However, they have to be moved to a disk pool when needed in order to process them. And since millions of files are involved, some sort of automated bookkeeping is needed.
The solution adopted at KLOE is based on a central relational database; each new file produces an entry in the database and each time the file is moved or copied from a point of the system to another, the corresponding data in the database are updated to keep track of the change.

In this paper the tables used to keep track of the files are described. Moreover, a short introduction about the way the files get produced and managed is also given.

Index

Introduction

KLOE is acquiring huge amounts of data. To reduce the complexity of managing all these data, they get stored as files and moved to tape for permanent storage. However, they have to be moved to a disk pool when needed in order to process them. And since millions of files are involved, some sort of automated bookkeeping is needed.

The solution adopted at KLOE is based on a central relational database; each new file produces an entry in the database and each time the file is moved or copied from a point of the system to another, the corresponding data in the database are updated to keep track of the change.

In this paper the tables used to keep track of the files are described. In the next sections there is also some information about the way the files get produced and managed in order to get a better understanding of the table fields.

The KLOE environment

The Data flow

To achieve its goals, KLOE is acquiring huge amounts of data; the estimated needs are around 200G events for a total storage size of 1 PB. To efficiently manage all these data, a multi−layer scheme is used.

Most of the data in the KLOE environment are produced as a result of the readout of the FEE[a] after the trigger system has identified a possibly interesting event. These data are collected together, formatted and written to a ***raw*** file on an ***online*** disk pool. Some events, like cosmic rays, are easy to identify[b] after they have been formatted and they are also very interesting for monitoring and calibration purposes. For these reasons they get duplicated during the data acquisition process and written to separate raw files that make another source of data. Whichever the source, after a raw file reaches a set maximum size, it is closed and marked *read−only*, *ready−to−be−analyzed* and *ready−to−be−archived.*

Another source of data are simulated or MC[c] events. They are produced by MC generators, that simulate physics dynamics and create events that contain the same information it would be read–out by the FEE (plus some MC specific data). These events are written to ***mcraw*** files on a temporary disk area. When the desired amount of events has been created, the file is closed and marked *read−only* and *ready−to−be−analyzed.*

Once a raw file is declared *ready−to−be−analyzed*, it is processed by an offline reconstruction process. Each event contained in the input file is reconstructed and assigned to one or more classes or streams. For each stream that was marked as to be saved, the event is written in a dedicated ***datarec*** file (*once for every stream*) on an ***offline*** disk pool. Once all the events from the input file has been processed, all the related datarec files are closed and marked *read−only* and *ready−to−be−archived.* A similar process takes place also when a mcraw file is declared *ready−to−be−analyzed*, but streaming is not performed; that is, all the events are written in a ***mc*** file. Moreover, when the mc file is closed, the mcraw file is deleted from its disk area, since all its data are contained in the new mc file[d].

---

[a] Front End Electronics

[b] or at least a big fraction of them

[c] Monte Carlo

[d] since streaming is not cutting out any data

When a file is declared *ready−to−be−archived*, it is processed by the archiver process that copies it to the tape library. Once archived, the file can be deleted from its disk pool to make space for new files.

When a raw, datarec or mc file is declared *read−only*, it can be read−out by an analysis process. However, the disk pool where the file was written may not be (and normally is not) visible by the target machine and moreover the file could have been already canceled from the disk pool, so a copy of it is created on a ***recall*** disk pool by the data handling system. Once the copy is created (or an existing copy located), the requesting process can access it. The copy is guaranteed to stay on that disk pool as long as at least one process is accessing it, but can be deleted from the disk pool immediately afterwards to make space for a copy of another file.

Organizing the data

KLOE data taking is not a continuous and stable process. Physics data acquisition periods are interleaved with development, test, calibration and idle periods. Moreover, the environment conditions are continuously changing, although most of the time very slowly.

To effectively treat these different states of the data acquisition, the data taking is divided into ***runs***. A ***run*** is a period of data taking of a single type, where all the relevant conditions are assumed to be stable, that i.e. for a physics run means stable beam and environment conditions.

Since most of the conditions are considered to be stable during a run, the run itself is one of the most important indexes of the data taking, so raw files[a] are grouped by run. Moreover, data acquisition description and configuration maps and calibration data are stored as being valid for a range of runs, but this topic is outside the scope of this document.

Another important parameter is the so−called **datarec version**, i.e. the version of the reconstruction executable used to reconstruct the raw files. Differently from the raw files, where a single copy of a set of files is present for a specific acquisition period, several datarec files can be produced from a single raw file by different versions of the reconstruction executable. This difference is due to the evolving nature of the reconstruction process and is also the reason why the raw files are preserved in the first place. Reconstructing and classifying is a hard and complex job; the methods are developed during the life of the experiment and, moreover, a small bug in the code can drastically change the result of the computation. For this reason, *fresh*[b] raw files get processed with the last version of the reconstruction executable and when there is some CPU time left, also the files analyzed with an obsolete datarec version get reprocessed.

A world aside is MC data generation. Since this source of data is not linked at all with the data acquisition, the run number is not used. Instead, the main parameter is the so−called **MC data card** used to generate the events, or better to say the **MC data card identifier**.

A MC data card contains the parameters that are given to the MC generation program that creates the mcraw file. At each iteration, known also as **MC run number**, a single mcraw file is produced. Several data cards has been created and new are added every time a new class of physics analysis has to be done.

---

[a] and consequentially also datarec files

[b] i.e. just acquired or not yet analyzed

The Database System

Two database systems are present in the KLOE environment:

1. a HepDB based system and
2. a RDBMS[a] based system.

The two database systems has been introduced in two different points in time and for different reasons. The first, HepDB based, system was introduced at the beginning of the offline development cycle and was chosen for its good integration in the offline software environment. Although it has proven to be well suited for the storage of reconstruction related data[b], it was judged not suitable for bookkeeping tasks. For this, and other, reasons a second, RDBMS based, database system has been put in place. In this paper, all the references to the KLOE database system are to be intended to the RDBMS based database system.

Data structure

Data in relational databases are organized as named tables.

- Each table has a well–defined name that cannot change in time.
- The columns of a table have a well–defined name and type. They are defined at table creation by the database manager and only minor modifications[c] are allowed to them during the life of the database.
- Rows can be inserted, updated and deleted at any time by the users. Moreover, the database system itself is free to change their order[d] at any time.

A table can be also thought to as an dynamic array of *C*–like structures, where each row is a line of the array and the columns are the fields of the structure. The array itself can be thought as bound to a global variable of fixed name.

Available types

Since each column has a well–defined type associated with it, also the list of available types is well–defined. The complete list of all the available types is however too long for this type of paper and moreover some of the available types are also RDBMS–vendor dependent. Instead, the list of types used inside the KLOE database is given:

Type	*Description*
integer	Integer number (like –836, 0, 9994234)
real	Real or floating point number (like –1.3E–3, 3.14, 0.0002, 45e97)
char(*size***)**	Fixed size string of *size* characters (**char(3)** could be 'f04', 'who', ' ')
varchar(*size***)**	Variable size string of up to *size* characters (**varchar(5)** could be 'table', 'me', 'who')
timestamp	Point in time, expressed also as a **char(26)** (like '2000–12–15–09.58.30.231032', '1999–01–31–23.21.00.001345', '2000–08–01–00.00.00.000000')

One peculiar characteristic of the relational databases is also the possibility to assign a NULL value to a field. The NULL value is different from any other value of the column type and can be used as a special case to mark unknown or not yet available data.

---

[a] Relational Database Management System
[b] i.e. detector geometry and calibration data
[c] essentially the introduction of new columns
[d] i.e. to improve the performances

Keys and constraints

A nice feature of RDBMS systems is the possibility to set constraints about the data; they can be used to prevent invalid or duplicate data to be inserted into the database.

Four types of constraints are used in our context:

Constraint	*Description*
NULL–able fields	List of columns that can contain NULL values. Can be used to prevent a NULL value to be assigned to a field.
Primary key	List of columns that are used to distinguish between rows. Two rows cannot have the same value for the set of fields associated with the primary key. The search for a row given the values of the primary key is usually the most efficient one.
Unique constraint	List of columns that are used to distinguish between rows. Similarly as with a primary key, two rows cannot have the same value for the set of fields associated with the unique constraint. The search using these fields is normally slightly less efficient compared to the use of the primary key.
Foreign key	This is a referential constraint. The values of the fields listed in the foreign key must match the values of a row in the foreign table in order to be valid. Foreign keys are used to prevent the insertion of invalid or meaningless data into the database and are also useful hints for the creation of joins.

Views of the data

Another nice feature of RDBMS systems is the possibility to create views of the data. A view is seen by the end–user as a table, made as usual of rows and columns. However, the data presented by a view is actually not stored inside the view, but is instead extracted from one or more tables each time the requested data is needed.

The views are created by the database manager to simplify the access to data that are stored in several tables. Splitting the data over different tables is indeed often needed to improve the performances and to guarantee the consistency of the data.

Accessing the data

The access to the data stored in a relational database system is achieved via the SQL[a] query language. The description of this language is however beyond the scope of this document.

---

[a] Structured Query Language

The database structure

The database can be classified using two different metrics:

1. distinction between base tables and extended views
2. classification of tables/views based on the kind of data they contain

The first classification affects the way the database is used. The base tables are organized in such a way that redundancy is minimized and consistency problems are avoided, which is very useful when new information has to be inserted into the database. The extended views are instead query oriented; they contain a lot of redundant information in order to simplify the queries.

The second classification is instead more structural; indeed not all the data inside the database contain the same kind of information. Two main categories of data can be found in the bookkeeping database:

- content oriented
- position oriented

The tables/views of the first kind contain information about runs, files, etc. themselves, while the tables/views of the second kind contain the position of those files on the disk areas. The big difference between the two categories is that the information about a file is stored exactly once in the first category, while in the second a file can be listed zero, one or more times, depending on how many copies of it are present on the disk areas.

The structure of this paper is organized to privilege the first classification to the second. This is done since most of the users most of the time do query or insertions and it is very rare that both activities are performed at the same time by a single user.

Anyhow, most of the users will probably be interested in doing queries. Those users are encouraged to concentrate on the **Extended views** paragraph, since the use of views is most of the time much easier that the use of base tables. However, a basic notion about the way the data are stored inside the basic tables can be anyhow useful since it can help to write more efficient queries.

Other data inside the KLOE database

This paper has the ambition to become the reference manual for the bookkeeping part of the KLOE database. In the same database are however present several other sets of data; at the time of writing the most important ones are:

Data set	***Description***
System Description[a]	This data set contains the description of the KLOE computing system, both from the point of view of available machines and disk areas. Note that some of the tables of this set are used as foreign tables of the foreign keys of the bookkeeping tables.
DAQ Configuration	This data set contains the configuration of the DAQ during the life of the experiment.

The complete description of the above data sets is obviously outside the scope of this document. The interested reader should consult the corresponding manuals for more information.

---

[a] see KLOE note 227

Base Tables

The KLOE bookkeeping database contains a lot of different information; there are data about DAQ runs, data about raw and datarec files, data about Monte Carlo data cards and data about mcraw and mc files, just to mention the most important ones. Moreover, several copies of a file can coexist at the same time in the KLOE computing environment and the bookkeeping database must keep track of all of them.

To efficiently manage all these different types of data and to minimize the redundancy of the data, several tables are used to achieve the task. Moreover, to guarantee the consistency of the inserted data, foreign keys are used whenever possible.

Since the number of involved tables is so large, to better describe them, this chapter is divided in two sections. The first is dedicated to the content related tables while the second treats position related ones.

The principal difference between the two classes of tables regards what type of information they treat. Content related tables contain information about runs, MC data cards and files; all information that is supposed to be added sequentially and never deleted. Position related tables contain instead information about the position of files on the disk areas; a kind of information that changes continuously in time. Moreover, since only the current situation is kept, entries can also be deleted from the tables when files get removed from disk.

Content related tables

The universe of tables described in this section is very heterogeneous. There is not a single way to catalogue them, but three of them will maybe do.

The first classification is based on how the information are entered in the tables. Using this method, two classes of tables can be found:

Class	*Description*	*Related tables*
Description	The tables present in this category are updated only by the database manager. The data contained in them get changed very rarely, if ever.	descript.archived_status descript.mccard_base descript.mccard_card descript.mccard_group descript.mcstream_descript descript.mcstream_group descript.reconstruct_descript descript.run_type_descript descript.stream_descript descript.stream_offline
Logger	The tables in this category are updated by the DAQ, reconstruction and MC processes. The data are updated relatively often and the most used operation is the insertion of new entries.	logger.datarec_logger logger.datarec_raws logger.datarec_datarec logger.datarec_base_raws logger.mc_runs logger.mc_logger logger.mc_streams logger.raw_logger logger.run_logger

The second classification is based on the source of data the tables are treating. With this method, three classes can be outlined:

Class	*Description*	*Related tables*
DAQ	The tables present in this class are connected to the data taken during the data acquisition.	logger.run_logger
		logger.raw_logger
		logger.datarec_logger
		logger.datarec_raws
		logger.datarec_datarec
		logger.datarec_base_raws
		descript.reconstruct_descript
		descript.run_type_descript
		descript.stream_descript
		descript.stream_offline
MC	The tables present in this class are connected to the data generated by the Monte Carlo generators.	logger.mc_runs
		logger.mc_logger
		logger.mc_streams
		descript.mccard_base
		descript.mccard_card
		descript.mccard_group
		descript.mcstream_descript
		descript.mcstream_group
Not connected	The table present in this class is used in relation with data coming both from DAQ and MC.	descript.archived_status

The third and last classification is based on the type of files it treats. As explained in the introduction, four types of files get produced; raw, datarec, mcraw and mc, but since the mcraw files are just an intermediate step for the creation of mc files, the two will be put together for what concerns this classification. However, the tables treated in this section do not treat only files, so another two categories are added; run related tables and common file tables.

raw	*datarec*	*mcraw & mc*
logger.raw_logger	logger.datarec_logger	logger.mc_runs
descript.stream_descript	logger.datarec_raws	logger.mc_logger
	logger.datarec_datarec	logger.mc_streams
	logger.datarec_base_raws	descript.mccard_base
	descript.stream_offline	descript.mccard_card
		descript.mccard_group
		descript.mcstream_descript
		descript.mcstream_group

run	*common*
logger.run_logger	descript.archived_status
descript.run_type_descript	
descript.reconstruct_descr	

This third classification is also the one used to group together the tables during this section; with run related tables first, followed by the file related tables and the common file related table put at the end.

logger.run_logger

This table contains the list of all the runs taken during the life of the experiment.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Nr	integer	no	Run number
Run_Type_Id	integer[a]	no	Run type
User_id	integer	no	Not significant at the moment
Start_Time	timestamp	no	Run active period End_Time can be used to check if a run is still active or not (NULL value).
End_Time	timestamp	yes	
Zero_Sup	integer (as boolean)	no	Was zero suppression in use?
On_Disk	integer (as boolean)	no	Any physics data produced during the run?
Total_Events	integer	yes	Number of physics events acquired during the run
Normal_End	integer (as boolean)	yes	Run ended on error? (*not of the value*)
DAFNE_Im	real	yes	DAΦNE beam currents (in mA) at start of run (Im – e^-, Ip – e^+)
DAFNE_Ip			
DAFNE_LTm	integer	yes	DAΦNE beam lifetimes (in secs) at start of run (LTm – e^-, LTp – e^+)
DAFNE_LTp			
DAFNE_TSW1	integer (32 bit)	yes	Bit pattern of the DAΦNE beam (bunch crossing) Use as a 128 bit value
DAFNE_TSW2			
DAFNE_TSW3			
DAFNE_TSW4			
DAFNE_Luminosity	real	yes	Not significant at the moment
DAFNE_Fill	integer	yes	DAΦNE fill number at start of run
DAFNE_Vacuum	real	yes	Vacuum in the KLOE interaction region at start of run
Magnet_Current	real	yes	Current (in A) in the KLOE magnet solenoid (at start of run)
L3_BhaBha	integer	yes	BhaBhas found during the run by the L3 filters and by the Trigger Monitors
TrgMon_BhaBha			
L3_Luminosity	real	yes	Average luminosity (in $cm^{-2}s^{-1}$) of the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Luminosity			

---

[a] foreign key of descript.run_type_descript

Field name	*Field type*	*Can be NULL?*	*Description*
L3_Int_Lum	real	yes	Luminosity (in cm^{-2} or $10^{-33}nb^{-1}$) integrated during the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Int_Lum			
Calib_Word	integer (32 bit)	no	Bit pattern indicating the current status of the calibration for the run
Reconstruct_Id	integer[a]	yes	Vote given by a human operator
Comment	varchar(40)	yes	Free text comment given by a human operator

Primary key:

- Run_Nr

Foreign keys:

Local field	*Foreign table*	*Foreign field*
Run_Type_Id	descript.run_type_descript	Run_Type_Id
Reconstruct_Id	descript.reconstruct_descr	Reconstruct_Id

---

[a] foreign key of descript.reconstruct_descr

descript.run_type_descript

This table contains all the available run types.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Type_Id	integer	no	Run type identifier, used for foreign keys
Code	varchar(8)	no	Run type code, used for human readable interaction
WaitSync	integer (as boolean)	no	Need the DAQ farm builder wait for the sync cycle before validating the arrived data?
Description	varchar(48)	no	Free text description

Primary key:

- Run_Type_Id

Unique constraint:

- Code

descript.reconstruct_descr

This table contains all the possible votes a human operator can assign to a run.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Reconstruct_Id	integer	no	Identifier, used for foreign keys
Description	varchar(32)	no	Free text description

Primary key:

- Reconstruct_Id

logger.raw_logger

This table contains the list of all the raw files produced during the data taking.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Nr	integer[a]	no	Run number
Farm_Id	char(3)	no	Acquisition machine identifier
Receiver_Nr	integer	no	Builder identifier
Builder_Nr			
Stream_Id	integer[b]	no	Stream identifier
Recorder_Nr	integer	no	Recorder number
Gb_Nr	integer	no	Sequential number
Filename	varchar(64)	no	File name
Creation_Time	timestamp	no	Creation time
Closure_Time	timestamp	yes	Closure time Can be used to find out if the file is still being written (NULL value)
First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the recorder.
Last			
Size	integer	yes	File size
Total_Events	integer	yes	Number of events inside the file
Archived	integer[c]	no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_Id	char(3) [d]	yes	Filespace where the file is archived.
Filespace_Id	integer [d]		
Analyzed	integer	yes	The highest version of datarec used to reconstruct this file (NULL if never reconstructed)

Primary key:

- Run_Nr
- Farm_Id
- Receiver_Nr
- Builder_Nr
- Stream_Id
- Recorder_Nr
- Gb_Nr

---

[a] foreign key of logger.run_logger

[b] foreign key of descript.stream_descript

[c] foreign key of descript.archived_status

[d] foreign key of descript.filespace_farm (together with Stream_Id)

Unique constraint:

- Filename

Foreign keys:

Local field(s)	***Foreign table***	***Foreign field(s)***
Run_Nr	logger.run_logger	Run_Nr
Stream_Id	descript.stream_descript	Stream_Id
Archived	descript.archived_status	Status
TServer_Id	descript.filespace_farm	TServer_Id
Stream_Id		Stream_Id
Filespace_Id		Filespace_Id

descript.stream_descript

This table contains the list of all the possible sources of raw files. Moreover, since not all the sources contain equally important data, there are also some flags that classify each source.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Stream_Id	integer	no	Stream identifier, used for foreign keys
Code	varchar(8)	no	Stream code, used for human readable interaction
Analyze	integer (as boolean)	no	If false, it will not be reconstructed.
All_Data	integer (as boolean)	no	If false, it could have been produced using only a subset of the available events (i.e. by a spy process).
Description	varchar(32)	no	Free text description

Primary key:

- Stream_Id

Unique constraint:

- Code

logger.datarec_logger

This table contains the list of all the datarec files produced during the reconstruction process.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Nr	integer[a]	no	Run number
Version	integer	no	Version of the reconstruction executable
Offline_Id	char(3)	no	Reconstruction machine identifier
Datarec_Nr	integer	no	Process identifier
Stream_Id	integer[b]	no	Stream identifier
Gb_Nr	integer	no	Sequential number
Filename	varchar(64)	no	File name
Creation_Time	timestamp	no	Creation time
Closure_Time	timestamp	yes	Closure time Can be used to find out if the file is still being written (NULL value)
First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the process.
Last			
Size	integer	yes	File size
Total_Events	integer	yes	Number of events inside the file
Archived	integer[c]	no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_Id	char(3)[d]	yes	Filespace where the file is archived.
Filespace_Id	integer		
Analyzed	integer	yes	Not used at the moment.

Primary key:

- Run_Nr
- Version
- Offline_Id
- Datarec_Nr
- Stream_Id
- Gb_Nr

Unique constraint:

- Filename

---

[a] foreign key of logger.run_logger
[b] foreign key of descript.stream_offline
[c] foreign key of descript.archived_status
[d] foreign key of descript.filespace_offline (together with Stream_Id)

Foreign keys:

Local field(s)	***Foreign table***	***Foreign field(s)***
Run_Nr	logger.run_logger	Run_Nr
Stream_Id	descript.stream_offline	Stream_Id
Archived	descript.archived_status	Status
TServer_Id	descript.filespace_offline	TServer_Id
Stream_Id		Stream_Id
Filespace_Id		Filespace_Id

descript.stream_offline

This table contains the list of all the possible datarec streams.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Stream_Id	integer	no	Stream identifier, used for foreign keys
Code	varchar(8)	no	Stream code, used for human readable interaction
Description	varchar(32)	no	Free text description

Primary key:

- Stream_Id

Unique constraint:

- Code

logger.datarec_raws

This table contains the relations between raw and datarec files of the datarec files that were reconstructed starting from raw files.

Fields description:

Field name	***Field type***		***Can be NULL?***	***Description***
Run_Nr	integer		no	Run number
Datarec_Version	integer	a	no	Version of the reconstruction executable
Datarec_Offline_Id	char(3)		no	Reconstruction machine identifier
Datarec_Nr	integer		no	Reconstruction process identifier
Datarec_Stream_Id	integer		no	Datarec stream identifier
Datarec_Gb_Nr	integer		no	Datarec file sequential number
Raw_Farm_Id	char(3)	b	no	Acquisition machine identifier
Raw_Receiver_Nr	integer		no	Builder identifier
Raw_Builder_Nr				
Raw_Stream_Id	integer		no	Raw stream identifier
Raw_Recorder_Nr	integer		no	Recorder number
Raw_Gb_Nr	integer		no	Raw file sequential number

Primary key:

All

Foreign keys:

Local field(s)	***Foreign table***	***Foreign field(s)***
Run_Nr	logger.datarec_logger	Run_Nr
Datarec_Version		Version
Datarec_Offline_Id		Offline_Id
Datarec_Nr		Datarec_Nr
Datarec_Stream_Id		Stream_Id
Datarec_Gb_Nr		Gb_Nr
Run_Nr	logger.raw_logger	Run_Nr
Raw_Farm_Id		Farm_Id
Raw_Receiver_Nr		Receiver_Nr
Raw_Builder_Nr		Builder_Nr
Raw_Stream_Id		Stream_Id
Raw_Recorder_Nr		Recorder_Nr
Raw_Gb_Nr		Gb_Nr

---

a foreign key of logger.datarec_logger (together with Run_Nr)

b foreign key of logger.raw_logger (together with Run_Nr)

logger.datarec_datarec

This table contains the relations between datarec files of the datarec files that were reconstructed starting from other datarec files.

Fields description:

Field name	*Field type*		*Can be NULL?*	*Description*
Run_Nr	integer		no	Run number
Datarec_Version	integer	[a]	no	Version of the reconstruction executable that created the output file
Datarec_Offline_Id	char(3)		no	Reconstruction machine identifier that created the output file
Datarec_Nr	integer		no	Reconstruction process identifier that created the output file
Datarec_Stream_Id	integer		no	Output datarec file stream identifier
Datarec_Gb_Nr	integer		no	Output datarec file sequential number
In_Dtr_Version	integer	[b]	no	Version of the reconstruction executable that created the input file
In_Dtr_Offline_Id	char(3)		no	Reconstruction machine identifier that created the input file
In_Dtr_Datarec_Nr	integer		no	Reconstruction process identifier that created the input file
In_Dtr_Stream_Id	integer		no	Input datarec file stream identifier
In_Dtr_Gb_Nr	integer		no	Input datarec file sequential number

Primary key:
All

Foreign keys:

Local field(s)	*Foreign table*	*Foreign field(s)*
Run_Nr	logger.datarec_logger	Run_Nr
Datarec_Version		Version
Datarec_Offline_Id		Offline_Id
Datarec_Nr		Datarec_Nr
Datarec_Stream_Id		Stream_Id
Datarec_Gb_Nr		Gb_Nr
Run_Nr	logger.datarec_logger	Run_Nr
In_Dtr_Version		Version
In_Dtr_Offline_Id		Offline_Id
In_Dtr_Datarec_Nr		Datarec_Nr
In_Dtr_Stream_Id		Stream_Id
In_Dtr_Gb_Nr		Gb_Nr

---

[a] foreign key of logger.datarec_logger (together with Run_Nr)

[b] foreign key of logger.datarec_logger (together with Run_Nr)

logger.datarec_base_raws

This table contains the relations between datarec files and the base raw files they were produced from. If a datarec file was produced from other datarec files, the base raw files are found following the path down to the leaves.

This table is automatically maintained by a set of triggers. Nobody (neither the database administrators) should change anything inside it.

Fields description:

Field name	***Field type***		***Can be NULL?***	***Description***
Run_Nr	integer		no	Run number
Datarec_Version	integer	[a]	no	Version of the reconstruction executable
Datarec_Offline_Id	char(3)		no	Reconstruction machine identifier
Datarec_Nr	integer		no	Reconstruction process identifier
Datarec_Stream_Id	integer		no	Datarec stream identifier
Datarec_Gb_Nr	integer		no	Datarec file sequential number
Raw_Farm_Id	char(3)	[b]	no	Acquisition machine identifier
Raw_Receiver_Nr	integer		no	Builder identifier
Raw_Builder_Nr				
Raw_Stream_Id	integer		no	Raw stream identifier
Raw_Recorder_Nr	integer		no	Recorder number
Raw_Gb_Nr	integer		no	Raw file sequential number

Primary key:

All

Foreign keys:

Local field(s)	***Foreign table***	***Foreign field(s)***
Run_Nr	logger.datarec_logger	Run_Nr
Datarec_Version		Version
Datarec_Offline_Id		Offline_Id
Datarec_Nr		Datarec_Nr
Datarec_Stream_Id		Stream_Id
Datarec_Gb_Nr		Gb_Nr
Run_Nr	logger.raw_logger	Run_Nr
Raw_Farm_Id		Farm_Id
Raw_Receiver_Nr		Receiver_Nr
Raw_Builder_Nr		Builder_Nr
Raw_Stream_Id		Stream_Id
Raw_Recorder_Nr		Recorder_Nr
Raw_Gb_Nr		Gb_Nr

---

[a] foreign key of logger.datarec_logger (together with Run_Nr)

[b] foreign key of logger.raw_logger (together with Run_Nr)

logger.mc_runs

This table contains information about the way mcraw files are produced by the Monte Carlo generators.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
MCCard_Id	integer[a]	no	MC card identifier
MCRun_Nr	integer	no	Run number relative to the MC card
Seed_Stream	integer	no	Identifier of the seed stream relative to the MC card
End_Seed1	integer (32 bit)	yes[b]	Seed to be used by the next generation on the same seed stream
End_Seed2			
Start_Time	timestamp	no	Generation start time
End_Time	timestamp	yes	Generation end time Can be used to find out if generation is still in progress (NULL value)
MC_Version	integer	no	Version of the Monte Carlo generator executable
Offline_Id	char(3)	no	Generator machine identifier
Total_Events	integer	yes	Numer of events produced
Nr_Trigs	integer	no	Number of triggers to generated by Monte Carlo

Primary key:

- MCCard_Id
- MCRun_Nr

Foreign key:

Local field	***Foreign table***	***Foreign field***
MCCard_Id	descript.mccard_base	MCCard_Id

---

[a] foreign key of descript.mccard_base

[b] can be NULL only if generation is still in progress

logger.mc_streams

This table contains the number of events produced for every physics stream of each mcraw file.

Fields description:

Field name	***Field type***		***Can be NULL?***	***Description***
MCCard_Id	integer	[a]	no	MC card identifier
MCRun_Nr	integer			Run number relative to the MC card
MCStream_Id	integer[b]		no	Physics stream identifier
Total_Events	integer		no	Number of events

Primary key:

- MCCard_Id
- MCRun_Nr
- MCStream_Id

Foreign keys:

Local field(s)	***Foreign table***	***Foreign field(s)***
MCCard_Id	logger.mc_runs	MCCard_Id
MCRun_Nr		MCRun_Nr
MCStream_Id	descript.mcstream_descript	MCStream_Id

---

[a] foreign key of logger.mc_runs

[b] foreign key of descript.mcstream_descript

logger.mc_logger

This table contains the list of all the mc files produced.

Fields description:

Field name	***Field type***		***Can be NULL?***	***Description***
MCCard_Id	integer	[a]	no	MC card identifier
MCRun_Nr	integer			Run number relative to the MC card
Datarec_Version	integer		no	Version of the reconstruction executable
Offline_Id	char(3)		no	Reconstruction machine identifier
Filename	varchar(64)		no	File name
Creation_Time	timestamp		no	Creation time
Closure_Time	timestamp		yes	Closure time Can be used to find out if the file is still being written (NULL value)
Size	integer		yes	File size
Archived	integer[b]		no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_Id	char(3)	[c]	yes	Filespace where the file is archived.
Filespace_Id	integer			
Analyzed	integer		yes	Not used at the moment.

Primary key:

- MCCard_Id
- MCRun_Nr

Unique constraint:

- Filename

Foreign keys:

Local field(s)	***Foreign table***	***Foreign field(s)***
MCCard_Id	logger.mc_runs	MCCard_Id
MCRun_Nr		MCRun_Nr
Archived	descript.archived_status	Status
TServer_Id	descript.filespace_mc	TServer_Id
Filespace_Id		Filespace_Id

---

[a] foreign key of logger.mc_runs

[b] foreign key of descript.archived_status

[c] foreign key of descript.filespace_mc

descript.mccard_base

This table contains the descriptions of the MC data cards used for the production of mcraw files. Each entry contains the identifier and the text description of a MC data card.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
MCCard_Id	integer	no	MC card identifier, used for foreign keys
Description	varchar(60)	no	MC card description, used for human readable interaction
MCCard_Group	integer[a]	no	MC group identifier

Primary key:

- MCCard_Id

Unique constraint:

- Description

Foreign key:

Local field	***Foreign table***	***Foreign field***
MCCard_Group	descript.mccard_group	MCCard_Group

descript.mccard_group

This table contains the list of MC data card categories by which the data cards are grouped.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
MCCard_Group	integer	no	MC group identifier, used for foreign keys
Description	varchar(60)	no	MC group description, used for human readable interaction

Primary key:

- MCCard_Group

Unique constraint:

- Description

---

[a] foreign key of descript.mccard_group

descript.mccard_card

This table contains the data of the MC data cards used for the production of mcraw files. All the history of MC data cards is present in this table.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
MCCard_Id	integer[a]	no	MC card identifier
Creation_Time	timestamp	no	Creation time of this instance of the card
Comment	varchar(62)	yes	Human readable comment about this instance of the card
Nr_Trigs	integer	no	Number of triggers to generate at each run
Card	varchar(40k)	no	The data card

Primary key:

- MCCard_Id
- Creation_Time

Foreign key:

Local field	***Foreign table***	***Foreign field***
MCCard_Id	descript.mccard_base	MCCard_Id

---

[a] foreign key of descript.mccard_base

descript.mcstream_descript

This table contains the list of all the physics streams that can be produced by the MC generators.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
MCStream_Id	integer	no	Physics stream identifier, used for foreign keys
Code	varchar(15)	no	Physics stream code, used for human readable interaction
Description	varchar(60)	no	Free text description
MCStream_Group	integer[a]	no	Physics stream group identifier

Primary key:

- MCStream_Id

Unique constraint:

- Code

Foreign key:

Local field	***Foreign table***	***Foreign field***
MCStream_Group	descript.mcstream_group	MCStream_Group

descript.mcstream_group

This table contains the list of physics groups by which the MC physics streams are grouped.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
MCStream_Group	integer	no	Physics group identifier, used for foreign keys
Code	varchar(15)	no	Physics group code, used for human readable interaction
Description	varchar(60)	no	Free text description

Primary key:

- MCStream_Group

Unique constraint:

- Code

---

[a] foreign key of descript.mcstream_group

descript.archived_status

This table contains a text description of the archiving states.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Status	integer	no	Status identifier, used for foreign keys
Description	varchar(32)	no	Free text description

Primary key:

- Status

Position related tables

The tables described in this section treats data about the position of files on the disk areas. Since there are four different kinds of files; raw, datarec, mcraw and mc, and 2 different types of disks; local and recall disks, a two dimensional classification of the available table comes natural:

	raw	***datarec***	***mcraw***	***mc***
local disk pool	status.raw_online	status.datarec_offline	status.mc_temp	status.mc_offline
recall disk pool	status.raw_recalled	status.datarec_recalled		status.mc_recalled

However, since this is also a 2D paper, a linear classification is needed in order to make a total order of the tables. The classification of choice is the one based on the kind of files.

status.raw_online

This table contains the list of raw files that are currently residing on the DAQ online disk pool.

Fields description:

Field name	***Field type***		***Can be NULL?***	***Description***
Run_Nr	integer	[a]	no	Run number
Farm_Id	char(3)			Acquisition machine identifier
Receiver_Nr	integer			Builder number
Builder_Nr				
Stream_Id	integer			Stream identifier
Recorder_Nr	integer			Recorder number
Gb_Nr	integer			Sequential number
FarmDir_id	integer[b]		no	Disk pool area identifier
Nr_Act_Acc	integer		no	Number of processes currently accessing this copy of the file

Primary key:

- Run_Nr
- Farm_Id
- Receiver_nr
- Builder_Nr
- Stream_Id
- Recorder_Nr

Foreign keys:

Local field(s)	***Foreign table***	***Foreign field(s)***
Run_Nr	logger.raw_logger	Run_Nr
Farm_Id		Farm_Id
Receiver_Nr		Receiver_Nr
Builder_Nr		Builder_Nr
Stream_Id		Stream_Id
Recorder_Nr		Recorder_Nr
Gb_Nr		Gb_Nr
Farm_Id	descript.farmdir_descript	Farm_Id
FarmDir_Id		FarmDir_Id

---

[a] foreign key of logger.raw_logger

[b] foreign key of descript.farmdir_descript (together with farm_id)

status.raw_recalled

This table contains the list of raw files that are currently residing on the recall disk areas.

Fields description:

Field name	***Field type***		***Can be NULL?***	***Description***
Run_Nr	integer	[a]	no	Run number
Farm_Id	char(3)			Acquisition machine identifier
Receiver_Nr	integer			Builder number
Builder_Nr				
Stream_Id	integer			Stream identifier
Recorder_Nr	integer			Recorder number
Gb_Nr	integer			Sequential number
DServer_Id	char(3)	[b]	no	Disk pool area identifier
DSrvDir_Id	integer			
Nr_Act_acc	integer		no	Number of processes currently accessing this copy of the file
Recalled_Time	timestamp		yes	Last access time If NULL, the file is not yet available on disk

Primary key:

- Run_Nr
- Farm_Id
- Receiver_Nr
- Builder_Nr
- Stream_Id
- Recorder_Nr
- Gb_Nr
- DServer_Id
- DSrvDir_Id

Foreign keys:

Local field(s)	***Foreign table***	***Foreign field(s)***
Run_Nr	logger.raw_logger	Run_Nr
Farm_Id		Farm_Id
Receiver_Nr		Receiver_Nr
Builder_Nr		Builder_Nr
Stream_Id		Stream_Id
Recorder_Nr		Recorder_Nr
Gb_Nr		Gb_Nr
DServer_Id	descript.dsrvdir_descript	DServer_Id
DSrvDir_Id		DSrvDir_Id

---

[a] foreign key of logger.raw_logger

[b] foreign key of descript.dsrvdir_descript

status.datarec_offline

This table contains the list of datarec files that are currently residing on the offline disk pool.

Fields description:

Field name	***Field type***		***Can be NULL?***	***Description***
Run_Nr	integer	a	no	Run number
Version	integer			Version of the reconstruction executable
Offline_Id	char(3)			Reconstruction machine identifier
Datarec_Nr	integer			Process identifier
Stream_Id	integer			Stream identifier
Gb_Nr	integer			Sequential number
OffDir_id	integer[b]		no	Disk pool area identifier
Nr_Act_Acc	integer		no	Number of processes currently accessing this copy of the file

Primary key:

- Run_Nr
- Version
- Offline_Id
- Datarec_Nr
- Stream_Id
- Gb_Nr

Foreign keys:

Local field(s)	***Foreign table***	***Foreign field(s)***
Run_Nr	logger.datarec_logger	Run_Nr
Version		Version
Offline_Id		Offline_Id
Datarec_Nr		Datarec_Nr
Stream_Id		Stream_Id
Gb_Nr		Gb_Nr
Offline_Id	descript.offdir_descript	Offline_Id
OffDir_Id		OffDir_Id

---

[a] foreign key of logger.datarec_logger

[b] foreign key of descript.offdir_descript (together with offline_id)

status.datarec_recalled

This table contains the list of datarec files that are currently residing on the recall disk areas.

Fields description:

Field name	***Field type***		***Can be NULL?***	***Description***
Run_Nr	integer	[a]	no	Run number
Version	integer			Version of the reconstruction executable
Offline_Id	char(3)			Reconstruction machine identifier
Datarec_Nr	integer			Process identifier
Stream_id	integer			Stream identifier
Gb_Nr	integer			Sequential number
DServer_Id	char(3)	[b]	no	Disk pool area identifier
DSrvDir_Id	integer			
Nr_Act_acc	integer		no	Number of processes currently accessing this copy of the file
Recalled_Time	timestamp		yes	Last access time If NULL, the file is not yet available on disk

Primary key:

- Run_Nr
- Version
- Offline_Id
- Datarec_Nr
- Stream_Id
- Gb_Nr
- DServer_Id
- DSrvDir_Id

Foreign keys:

Local field(s)	***Foreign table***	***Foreign field(s)***
Run_Nr	logger.datarec_logger	Run_Nr
Version		Version
Offline_Id		Offline_Id
Datarec_Nr		Datarec_Nr
Stream_Id		Stream_Id
Gb_Nr		Gb_Nr
DServer_Id	descript.dsrvdir_descript	DServer_Id
DSrvDir_Id		DSrvDir_Id

---

[a] foreign key of logger.datarec_logger

[b] foreign key of descript.dsrvdir_descript

status.mc_temp

This table contains the list of mcraw files and their position on local disks.

Fields description:

Field name	***Field type***		***Can be NULL?***	***Description***
MCCard_Id	integer	[b]	no	MC card identifier
MCRun_Nr	integer			Run number relative to the MC card
Offline_Id	char(3)[a]			Generator machine identifier
Filepath	varchar(240)		no	File path (directory + file name)

Primary key:

- MCCard_Id
- MCRun_Nr

Unique constraint:

- Offline_Id
- Filepath

Foreign keys:

Local field(s)	***Foreign table***	***Foreign field(s)***
MCCard_Id	logger.mc_runs	MCCard_Id
MCRun_Nr		MCRun_Nr
Offline_id		Offline_Id
Offline_Id	descript.mc_node	Offline_Id

---

[a] foreign key of descript.mc_node

[b] foreign key of logger.mc_runs

status.mc_offline

This table contains the list of mc files that are currently residing on the offline disk pool.

Fields description:

Field name	***Field type***		***Can be NULL?***	***Description***
MCCard_id	integer	[a]	no	MC card identifier
MCRun_Nr	integer			Run number relative to the MC card
Offline_Id	char(3)			Reconstruction machine identifier
OffDir_id	integer[b]		no	Disk pool area identifier
Nr_Act_Acc	integer		no	Number of processes currently accessing this copy of the file

Primary key:

- MCCard_Id
- MCRun_Nr

Foreign keys:

Local field(s)	***Foreign table***	***Foreign field(s)***
MCCard_Id	logger.mc_logger	MCCard_Id
MCRun_Id		MCRun_Id
Offline_id		Offline_id
Offline_Id	descript.offdir_descript	Offline_Id
OffDir_Id		OffDir_Id

---

[a] foreign key of logger.mc_logger

[b] foreign key of descript.offdir_descript (together with offline_id)

status.mc_recalled

This table contains the list of mc files that are currently residing on the recall disk areas.

Fields description:

Field name	*Field type*		*Can be NULL?*	*Description*
MCCard_Id	integer	[a]	no	MC card identifier
MCRun_Nr	integer			Run number relative to the MC card
DServer_Id	char(3)	[b]	no	Disk pool area identifier
DSrvDir_Id	integer			
Nr_Act_acc	integer		no	Number of processes currently accessing this copy of the file
Recalled_Time	timestamp		yes	Last access time If NULL, the file is not yet available on disk

Primary key:

- MCCard_Id
- MCRun_Nr
- DServer_Id
- DSrvDir_Id

Foreign keys:

Local field(s)	*Foreign table*	*Foreign field(s)*
MCCard_Id	logger.mc_logger	MCCard_Id
MCRun_Nr		MCRun_Nr
DServer_Id	descript.dsrvdir_descript	DServer_Id
DSrvDir_Id		DSrvDir_Id

---

[a] foreign key of logger.mc_logger

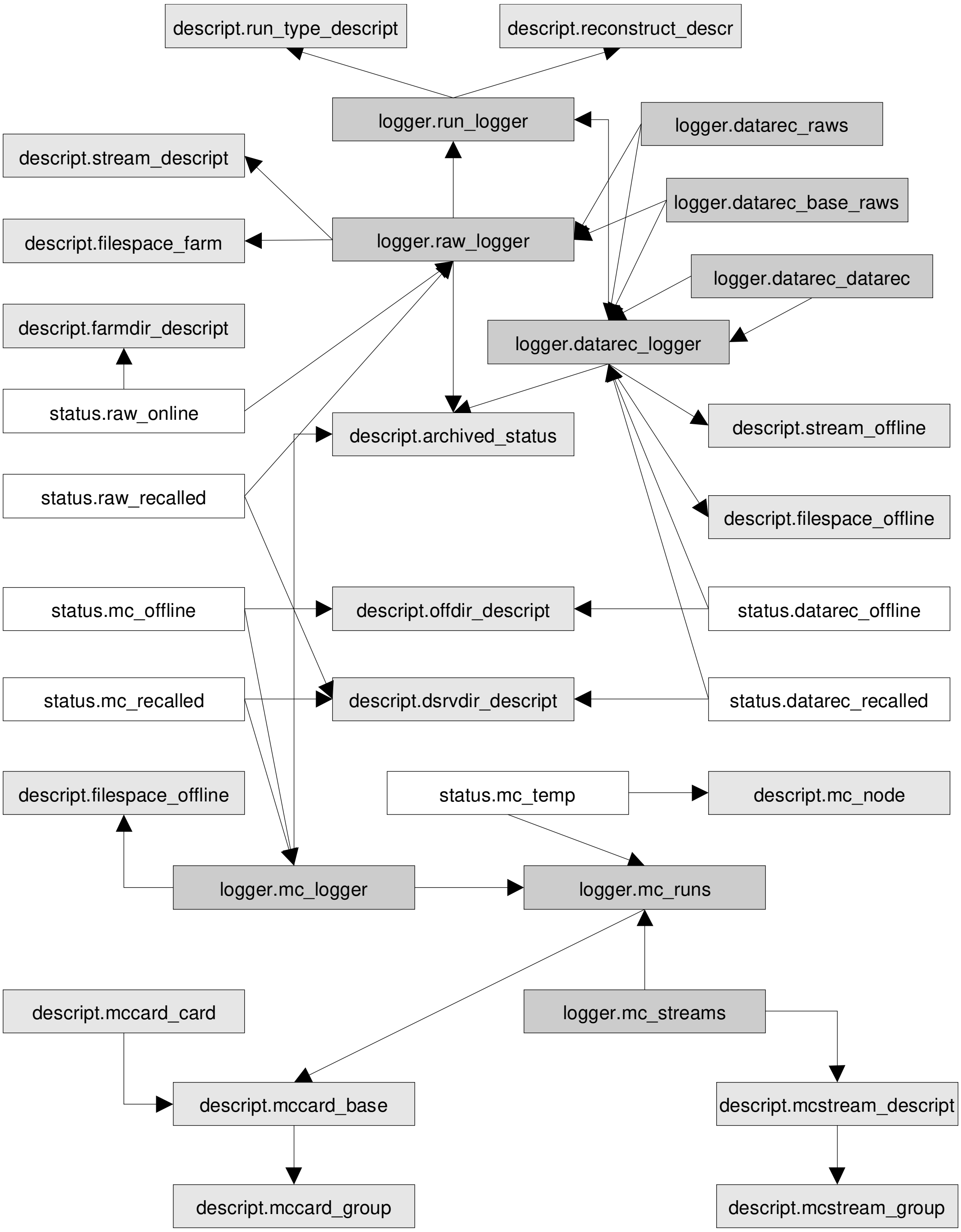
[b] foreign key of descript.dsrvdir_descript

Summary of tables

Table Name	*Description*
descript.archived_status	This table contains a text description of the archiving states.
descript.mccard_base	This table contains the descriptions of the MC data cards used for the production of mcraw files. Each entry contains the identifier and the text description of a MC data card.
descript.mccard_card	This table contains the data of the MC data cards used for the production of mcraw files. All the history of MC data cards is present in this table.
descript.mccard_group	This table contains the list of MC data card categories by which the data cards are grouped.
descript.mcstream_descript	This table contains the list of all the physics streams that can be produced by the MC generators.
descript.mcstream_group	This table contains the list of physics groups by which the MC physics streams are grouped.
descript.reconstruct_descript	This table contains all the possible votes a human operator can assign to a run.
descript.run_type_descript	This table contains all the available run types.
descript.stream_descript	This table contains the list of all the possible sources of raw files. Moreover, since not all the sources contain equally important data, there are also some flags that classify each source.
descript.stream_offline	This table contains the list of all the possible datarec streams.
logger.datarec_logger	This table contains the list of all the datarec files produced during the reconstruction process.
logger.datarec_raws	This table contains the relations between raw and datarec files of the datarec files that were reconstructed starting from raw files.
logger.datarec_datarec	This table contains the relations between datarec files of the datarec files that were reconstructed starting from other datarec files.
logger.datarec_base_raws	This table contains the relations between datarec files and the base raw files they were produced from. If a datarec file was produced from other datarec files, the base raw files are found following the path down to the leaves.
logger.mc_runs	This table contains information about the way mcraw files are produced by the Monte Carlo generators.
logger.mc_logger	This table contains the list of all the mc files produced.
logger.mc_streams	This table contains the number of events produced for every physics stream of each mcraw file.
logger.raw_logger	This table contains the list of all the raw files produced during the data taking.
logger.run_logger	This table contains the list of all the runs taken during the life of the experiment.
status.datarec_offline	This table contains the list of datarec files that are currently residing on the offline disk pool.

Table Name	***Description***
status.datarec_recalled	This table contains the list of datarec files that are currently residing on the recall disk areas.
status.mc_temp	This table contains the list of mcraw files and their position on local disks.
status.mc_offline	This table contains the list of mc files that are currently residing on the offline disk pool.
status.mc_recalled	This table contains the list of mc files that are currently residing on the recall disk areas.
status.raw_online	This table contains the list of raw files that are currently residing on the DAQ online disk pool.
status.raw_recalled	This table contains the list of raw files that are currently residing on the recall disk areas.

Foreign keys overview



Extended views

The base tables introduced in the previous section contain all the data needed to find the desired information about runs, files and their position on disk and tape. However, to avoid redundancy, several fields are stored only in a subset of tables and a join operation is needed[a] in order to build the complete information.

To simplify the query building, several views has been defined. They contain the joined information of the base tables, so the user can use a simple query to get all the fields he (or she) is interested in.

Being the views based on the base tables, they are grouped in the same way; content related and position related views.

The principal difference between the two classes of views regards what type of information they treat. Content related tables contain information about runs and files; all information that is supposed to be added sequentially and never deleted. Position related tables contain instead information about the position of files on the disk areas; a kind of information that changes continuously in time. Moreover, since only the current situation is kept, entries can also be deleted when files get removed from disk.

Content related views

The views described in this section can be divided in two classes, based on the source of data the tables are treating:

Class	*Description*	*Related views*
DAQ	The views present in this class are connected to the data taken during the data acquisition.	logger.run_data
		logger.raw_data
		logger.datarec_data
		logger.dtr_raws_data
		logger.dtr_dtr_data
		logger.dtr_base_raws_data
MC	The views present in this class are connected to the data generated by the Monte Carlo generators.	descript.mccard_descript
		logger.mc_runs_data
		logger.mc_runstreams_data
		logger.mc_data
		logger.mc_streams_data

Another classification can be made based on the type of files the view is managing; raw datarec, mcraw or mc, plus two additional classes of views that treat DAQ runs and Monte Carlo data cards:

run	***raw***	***datarec***
logger.run_data	logger.raw_data	logger.datarec_data logger.dtr_raws_data logger.dtr_dtr_data
		logger.dtr_base_raws_data
MC data card	***mcraw***	***mc***
descript.mccard_descript	logger.mc_runs_data logger.mc_runstreams_data	logger.mc_data logger.mc_streams_data

---

[a] and often even more than one

The second classification can actually be included in the first one, so in this section the views are ordered following both classifications at the same time.

logger.run_data

This view contains the list of runs taken during the life of the experiment.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Nr	integer	no	Run number
Run_Type_Id	integer	no	Run type identifier; numeric and human readable
Run_Type_Code	varchar(8)		
WaitSync	integer (as boolean)	no	Need the DAQ farm builder wait for the sync cycle before validating the arrived data?
User_id	integer	no	Not significant at the moment
Start_Time	timestamp	no	Run active period End_Time can be used to check if a run is still active or not (NULL value).
End_Time	timestamp	yes	
Zero_Sup	integer (as boolean)	no	Was zero suppression in use?
On_Disk	integer (as boolean)	no	Any physics data produced during the run?
Total_Events	integer	yes	Number of physics events acquired during the run
Torta_Name	varchar(16)	yes	Name of the TORTA configuration used during the run
Normal_End	integer (as boolean)	yes	Run ended on error? (*not of the value*)
DAFNE_Im	real	yes	DAΦNE beam currents (in mA) at start of run (Im – e^-, Ip – e^+)
DAFNE_Ip			
DAFNE_LTm	integer	yes	DAΦNE beam lifetimes (in secs) at start of run (LTm – e^-, LTp – e^+)
DAFNE_LTp			
DAFNE_TSW1	integer (32 bit)	yes	Bit pattern of the DAΦNE beam (bunch crossing) Use as a 128 bit value
DAFNE_TSW2			
DAFNE_TSW3			
DAFNE_TSW4			
DAFNE_Luminosity	real	yes	Not significant at the moment
DAFNE_Fill	integer	yes	DAΦNE fill number at start of run
DAFNE_Vacuum	real	yes	Vacuum in the KLOE interaction region at start of run
Magnet_Current	real	yes	Current (in A) in the KLOE magnet solenoid (at start of run)

Field name	***Field type***	***Can be NULL?***	***Description***
L3_BhaBha TrgMon_BhaBha	integer	yes	BhaBhas found during the run by the L3 filters and by the Trigger Monitors
L3_Luminosity TrgMon_Luminosity	real	yes	Average luminosity (in $cm^{-2}s^{-1}$) of the run, estimated by the L3 filters and the Trigger Monitors
L3_Int_Lum TrgMon_Int_Lum	real	yes	Luminosity (in cm^{-2} or $10^{-33}nb^{-1}$) integrated during the run, estimated by the L3 filters and the Trigger Monitors
Calib_Word	integer (32 bit)	no	Bit pattern indicating the current status of the calibration for the run
Reconstruct_Id	integer	yes	Vote given by a human operator
Comment	varchar(40)	yes	Comment given by a human operator

Primary key:

- Run_Nr

Tables used:

- logger.run_logger
- descript.run_type_descript
- kloemaps.torta_map[a]

---

[a] crossref.torta_run also used to speed–up the join

logger.raw_data

This view contains the list of all the raw files produced during the data taking. It contains both the data about the files themselves and the data about the runs they belong to.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Nr	integer	no	Run number
Farm_Id	char(3)	no	Acquisition machine identifier
Receiver_Nr	integer	no	Builder identifier
Builder_Nr			
Stream_Id	integer	no	Stream identifier; numeric and human readable
Stream_Code	varchar(8)		
Stream_Analyze	integer (as boolean)	no	If false, the file will not be reconstructed.
Stream_AllData	integer (as boolean)	no	If false, the file could have been produced using only a subset of the available events (i.e. by a spy process).
Recorder_Nr	integer	no	Recorder number
Gb_Nr	integer	no	Sequential number
Filename	varchar(64)	no	File name
Run_Type_Id	integer	no	Run type identifier; numeric and human readable
Run_Type_Code	varchar(8)		
WaitSync	integer (as boolean)	no	Need the DAQ farm builder wait for the sync cycle before validating the arrived data?
Run_Start_Time	timestamp	no	Run active period Run_End_Time can be used to check if a run is still active or not (NULL value).
Run_End_Time	timestamp	yes	
Zero_Sup	integer (as boolean)	no	Was zero suppression in use?
Run_On_Disk	integer (as boolean)	no	Any physics data produced during the run?
Run_Events	integer	yes	Number of physics events acquired during the run
Torta_Name	varchar(16)	yes	Name of the TORTA configuration used during the run
Run_Normal_End	integer (as boolean)	yes	Run ended on error? (*not of the value*)
DAFNE_Im	real	yes	DAΦNE beam currents (in mA) at start of run (Im – e^-, Ip – e^+)
DAFNE_Ip			

Field name	***Field type***	***Can be NULL?***	***Description***
DAFNE_LTm	integer	yes	DAΦNE beam lifetimes (in secs) at start of run (LTm – e^-, LTp – e^+)
DAFNE_LTp			
DAFNE_TSW1	integer (32 bit)	yes	Bit pattern of the DAΦNE beam (bunch crossing) Use as a 128 bit value
DAFNE_TSW2			
DAFNE_TSW3			
DAFNE_TSW4			
DAFNE_Fill	integer	yes	DAΦNE fill number at start of run
DAFNE_Vacuum	real	yes	Vacuum in the KLOE interaction region at start of run
Magnet_Current	real	yes	Current (in A) in the KLOE magnet solenoid (at start of run)
L3_BhaBha	integer	yes	BhaBhas found during the run by the L3 filters and by the Trigger Monitors
TrgMon_BhaBha			
L3_Luminosity	real	yes	Average luminosity (in $cm^{-2}s^{-1}$) of the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Luminosity			
L3_Int_Lum	real	yes	Luminosity (in cm^{-2} or $10^{-33}nb^{-1}$) integrated during the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Int_Lum			
Calib_Word	integer (32 bit)	no	Bit pattern indicating the current status of the calibration for the run
Reconstruct_Id	integer	yes	Vote given by a human operator
Run_Comment	varchar(40)	yes	Comment given by a human operator
Creation_Time	timestamp	no	Creation time
Closure_Time	timestamp	yes	Closure time Can be used to find out if the file is still being written (NULL value)
First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the recorder.
Last			
Size	integer	yes	File size
Total_Events	integer	yes	Number of events inside the file
Archived	integer	no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_id	char(3)	yes	Filespace where the file is archived.
Filespace_Id	integer		
Analyzed	integer	yes	The highest version of datarec used to reconstruct this file (NULL if never reconstructed)
Nr_recalled	integer	no	Number of copies on user recall areas

Primary key:

- Run_Nr
- Farm_Id
- Receiver_Nr
- Builder_Nr
- Stream_Id
- Recorder_Nr
- Gb_Nr

Unique constraint:

- Filename

Tables and views used:

- logger.raw_logger
- *logger.run_data*
    - logger.run_logger
    - descript.run_type_descript
    - kloemaps.torta_map
- descript.stream_descript

logger.datarec_data

This view contains the list of all the datarec files produced during the reconstruction process. It contains both the data about the files themselves and the data about the runs they belong to.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Nr	integer	no	Run number
Version	integer	no	Version of the reconstruction executable
Offline_Id	char(3)	no	Reconstruction machine identifier
Datarec_Nr	integer	no	Process identifier
Stream_Id	integer	no	Stream identifier; numeric and human readable
Stream_Code	varchar(8)		
Gb_Nr	integer	no	Sequential number
Filename	varchar(64)	no	File name
Run_Type_Id	integer	no	Run type identifier; numeric and human readable
Run_Type_Code	varchar(8)		
WaitSync	integer (as boolean)	no	Need the DAQ farm builder wait for the sync cycle before validating the arrived data?
Run_Start_Time	timestamp	no	Run active period Run_End_Time can be used to check if a run is still active or not (NULL value).
Run_End_Time	timestamp	yes	
Zero_Sup	integer (as boolean)	no	Was zero suppression in use?
Run_On_Disk	integer (as boolean)	no	Any physics data produced during the run?
Run_Events	integer	yes	Number of physics events acquired during the run
Torta_Name	varchar(16)	yes	Name of the TORTA configuration used during the run
Run_Normal_End	integer (as boolean)	yes	Run ended on error? (*not of the value*)
DAFNE_Im	real	yes	DAΦNE beam currents (in mA) at start of run (Im – e^-, Ip – e^+)
DAFNE_Ip			
DAFNE_LTm	integer	yes	DAΦNE beam lifetimes (in secs) at start of run (LTm – e^-, LTp – e^+)
DAFNE_LTp			
DAFNE_TSW1	integer (32 bit)	yes	Bit pattern of the DAΦNE beam (bunch crossing) Use as a 128 bit value
DAFNE_TSW2			
DAFNE_TSW3			
DAFNE_TSW4			

Field name	***Field type***	***Can be NULL?***	***Description***
DAFNE_Fill	integer	yes	DAΦNE fill number at start of run
DAFNE_Vacuum	real	yes	Vacuum in the KLOE interaction region at start of run
Magnet_Current	real	yes	Current (in A) in the KLOE magnet solenoid (at start of run)
L3_BhaBha	integer	yes	BhaBhas found during the run by the L3 filters and by the Trigger Monitors
TrgMon_BhaBha			
L3_Luminosity	real	yes	Average luminosity (in cm^{-2}s^{-1}) of the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Luminosity			
L3_Int_Lum	real	yes	Luminosity (in cm^{-2} or 10^{-33}nb^{-1}) integrated during the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Int_Lum			
Calib_Word	integer (32 bit)	no	Bit pattern indicating the current status of the calibration for the run
Reconstruct_Id	integer	yes	Vote given by a human operator
Run_Comment	varchar(40)	yes	Comment given by a human operator
Creation_Time	timestamp	no	Creation time
Closure_Time	timestamp	yes	Closure time Can be used to find out if the file is still being written (NULL value)
First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the process.
Last			
Size	integer	yes	File size
Total_Events	integer	yes	Number of events inside the file
Archived	integer	no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_id	char(3)	yes	Filespace where the file is archived.
Filespace_Id	integer		
Analyzed	integer	yes	Not used at the moment.
Nr_recalled	integer	no	Number of copies on user recall areas

Primary key:

- Run_Nr
- Version
- Offline_Id
- Datarec_Nr
- Stream_Id
- Gb_Nr

Unique constraint:

- Filename

Tables and views used:

- logger.datarec_logger
- *logger.run_data*
  - logger.run_logger
  - descript.run_type_descript
  - kloemaps.torta_map
- descript.stream_offline

logger.dtr_raws_data

This view contains the relations between raw and datarec files of the datarec files that were reconstructed starting from raw files. It contains information about the resulting datarec files, information about the source raw files and information about the runs they belong to.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Nr	integer	no	Run number
Run_Type_Id	integer	no	Run type identifier; numeric and human readable
Run_Type_Code	varchar(8)		
WaitSync	integer (as boolean)	no	Need the DAQ farm builder wait for the sync cycle before validating the arrived data?
Run_Start_Time	timestamp	no	Run active period Run_End_Time can be used to check if a run is still active or not (NULL value).
Run_End_Time	timestamp	yes	
Zero_Sup	integer (as boolean)	no	Was zero suppression in use?
Run_On_Disk	integer (as boolean)	no	Any physics data produced during the run?
Run_Events	integer	yes	Number of physics events acquired during the run
Torta_Name	varchar(16)	yes	Name of the TORTA configuration used during the run
Run_Normal_End	integer (as boolean)	yes	Run ended on error? (*not of the value*)
DAFNE_Im	real	yes	DAΦNE beam currents (in mA) at start of run (Im – e^-, Ip – e^+)
DAFNE_Ip			
DAFNE_LTm	integer	yes	DAΦNE beam lifetimes (in secs) at start of run (LTm – e^-, LTp – e^+)
DAFNE_LTp			
DAFNE_TSW1	integer (32 bit)	yes	Bit pattern of the DAΦNE beam (bunch crossing) Use as a 128 bit value
DAFNE_TSW2			
DAFNE_TSW3			
DAFNE_TSW4			
DAFNE_Fill	integer	yes	DAΦNE fill number at start of run
DAFNE_Vacuum	real	yes	Vacuum in the KLOE interaction region at start of run
Magnet_Current	real	yes	Current (in A) in the KLOE magnet solenoid (at start of run)
L3_BhaBha	integer	yes	BhaBhas found during the run by the L3 filters and by the Trigger Monitors
TrgMon_BhaBha			

Field name	***Field type***	***Can be NULL?***	***Description***
L3_Luminosity	real	yes	Average luminosity (in cm^{-2}s^{-1}) of the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Luminosity			
L3_Int_Lum	real	yes	Luminosity (in cm^{-2} or 10^{-33}nb^{-1}) integrated during the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Int_Lum			
Calib_Word	integer (32 bit)	no	Bit pattern indicating the current status of the calibration for the run
Reconstruct_Id	integer	yes	Vote given by a human operator
Run_Comment	varchar(40)	yes	Comment given by a human operator
DTR_Version	integer	no	Version of the reconstruction executable
DTR_Offline_Id	char(3)	no	Reconstruction machine identifier
DTR_Datarec_Nr	integer	no	Reconstruction process identifier
DTR_Stream_Id	integer	no	Datarec file stream identifier; numeric and human readable
DTR_Stream_Code	varchar(8)		
DTR_Gb_Nr	integer	no	Datarec file sequential number
DTR_Filename	varchar(64)	no	Datarec file name
DTR_Creation_Time	timestamp	no	Datarec file creation time
DTR_Closure_Time	timestamp	yes	Datarec file closure time Can be used to find out if the file is still being written (NULL value)
DTR_First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the reconstruction process.
DTR_Last			
DTR_Size	integer	yes	Datarec file size
DTR_Total_Events	integer	yes	Number of events inside the datarec file
DTR_Archived	integer	no	Is the datarec file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
DTR_TServer_id	char(3)	yes	Filespace where the datarec file is archived.
DTR_Filespace_Id	integer		
DTR_Analyzed	integer	yes	Not used at the moment.
Raw_Farm_Id	char(3)	no	Acquisition machine identifier
Raw_Receiver_Nr	integer	no	Builder identifier
Raw_Builder_Nr			
Raw_Stream_Id	integer	no	Raw stream identifier; numeric and human readable
Raw_Stream_Code	varchar(8)		
Raw_Stream_Analyze	integer (as boolean)	no	If false, the raw file should not have been reconstructed.

Field name	*Field type*	*Can be NULL?*	*Description*
Raw_Stream_AllData	integer (as boolean)	no	If false, the raw file could have been produced using only a subset of the available events (i.e. by a spy process).
Raw_Recorder_Nr	integer	no	Recorder number
Raw_Gb_Nr	integer	no	Raw file sequential number
Raw_Filename	varchar(64)	no	Raw file name
Raw_Creation_Time	timestamp	no	Raw file creation time
Raw_Closure_Time	timestamp	yes	Raw file closure time Can be used to find out if the file is still being written (NULL value)
Raw_First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the recorder.
Raw_Last			
Raw_Size	integer	yes	Raw file size
Raw_Total_Events	integer	yes	Number of events inside the raw file
Raw_Archived	integer	no	Is the raw file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
Raw_TServer_id	char(3)	yes	Filespace where the raw file is archived.
Raw_Filespace_Id	integer		
Raw_Analyzed	integer	yes	The highest version of datarec used to reconstruct the raw file (can be NULL if the raw file was never fully reconstructed)

Primary key:

- Run_Nr
- DTR_Version
- DTR_Offline_Id
- DTR_Stream_Id
- DTR_Datarec_Nr
- DTR_Gb_Nr
- Raw_Farm_Id
- Raw_Receiver_Nr
- Raw_Builder_Nr
- Raw_Stream_Id
- Raw_Recorder_Nr
- Raw_Gb_Nr

Unique constraint:

- DTR_Filename
- Raw_Filename

Tables and views used:

- logger.datarec_raws
- logger.datarec_logger
- logger.raw_logger
- *logger.run_data*
  - logger.run_logger
  - descript.run_type_descript
  - kloemaps.torta_map
- descript.stream_descript
- descript.stream_offline

logger.dtr_dtr_data

This view contains the relations between datarec files of the datarec files that were reconstructed starting from other datarec files. It contains information about both the source and the resulting datarec files and information about the runs they belong to.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Nr	integer	no	Run number
Run_Type_Id	integer	no	Run type identifier; numeric and human readable
Run_Type_Code	varchar(8)		
WaitSync	integer (as boolean)	no	Need the DAQ farm builder wait for the sync cycle before validating the arrived data?
Run_Start_Time	timestamp	no	Run active period Run_End_Time can be used to check if a run is still active or not (NULL value).
Run_End_Time	timestamp	yes	
Zero_Sup	integer (as boolean)	no	Was zero suppression in use?
Run_On_Disk	integer (as boolean)	no	Any physics data produced during the run?
Run_Events	integer	yes	Number of physics events acquired during the run
Torta_Name	varchar(16)	yes	Name of the TORTA configuration used during the run
Run_Normal_End	integer (as boolean)	yes	Run ended on error? (*not of the value*)
DAFNE_Im	real	yes	DAΦNE beam currents (in mA) at start of run (Im – e^-, Ip – e^+)
DAFNE_Ip			
DAFNE_LTm	integer	yes	DAΦNE beam lifetimes (in secs) at start of run (LTm – e^-, LTp – e^+)
DAFNE_LTp			
DAFNE_TSW1	integer (32 bit)	yes	Bit pattern of the DAΦNE beam (bunch crossing) Use as a 128 bit value
DAFNE_TSW2			
DAFNE_TSW3			
DAFNE_TSW4			
DAFNE_Fill	integer	yes	DAΦNE fill number at start of run
DAFNE_Vacuum	real	yes	Vacuum in the KLOE interaction region at start of run
Magnet_Current	real	yes	Current (in A) in the KLOE magnet solenoid (at start of run)
L3_BhaBha	integer	yes	BhaBhas found during the run by the L3 filters and by the Trigger Monitors
TrgMon_BhaBha			

Field name	***Field type***	***Can be NULL?***	***Description***
L3_Luminosity	real	yes	Average luminosity (in $cm^{-2}s^{-1}$) of the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Luminosity			
L3_Int_Lum	real	yes	Luminosity (in cm^{-2} or $10^{-33}nb^{-1}$) integrated during the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Int_Lum			
Calib_Word	integer (32 bit)	no	Bit pattern indicating the current status of the calibration for the run
Reconstruct_Id	integer	yes	Vote given by a human operator
Run_Comment	varchar(40)	yes	Comment given by a human operator
DTR_Version	integer	no	Version of the reconstruction executable that created the output file
DTR_Offline_Id	char(3)	no	Reconstruction machine identifier that created the output file
DTR_Datarec_Nr	integer	no	Reconstruction process identifier that created the output file
DTR_Stream_Id	integer	no	Output datarec file stream identifier; numeric and human readable
DTR_Stream_Code	varchar(8)		
DTR_Gb_Nr	integer	no	Output datarec file sequential number
DTR_Filename	varchar(64)	no	Output datarec file name
DTR_Creation_Time	timestamp	no	Output datarec file creation time
DTR_Closure_Time	timestamp	yes	Output datarec file closure time Can be used to find out if the file is still being written (NULL value)
DTR_First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the reconstruction process that created the output file.
DTR_Last			
DTR_Size	integer	yes	Output datarec file size
DTR_Total_Events	integer	yes	Number of events inside the output datarec file
DTR_Archived	integer	no	Is the output datarec file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
DTR_TServer_id	char(3)	yes	Filespace where the output datarec file is archived.
DTR_Filespace_Id	integer		
DTR_Analyzed	integer	yes	Not used at the moment.
InD_Version	integer	no	Version of the reconstruction executable that created the input file
InD_Offline_Id	char(3)	no	Reconstruction machine identifier that created the input file

Field name	*Field type*	*Can be NULL?*	*Description*
InD_Datarec_Nr	integer	no	Reconstruction process identifier that created the input file
InD_Stream_Id	integer	no	Input datarec file stream identifier; numeric and human readable
InD_Stream_Code	varchar(8)		
InD_Gb_Nr	integer	no	Input datarec file sequential number
InD_Filename	varchar(64)	no	Input datarec file name
InD_Creation_Time	timestamp	no	Input datarec file creation time
InD_Closure_Time	timestamp	yes	Input datarec file closure time Can be used to find out if the file is still being written (NULL value)
InD_First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the reconstruction process that created the input file.
InD_Last			
InD_Size	integer	yes	Input datarec file size
InD_Total_Events	integer	yes	Number of events inside the input datarec file
InD_Archived	integer	no	Is the input datarec file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
InD_TServer_id	char(3)	yes	Filespace where the input datarec file is archived.
InD_Filespace_Id	integer		
InD_Analyzed	integer	yes	Not used at the moment.

Primary key:

- Run_Nr
- DTR_Version
- DTR_Offline_Id
- DTR_Stream_Id
- DTR_Datarec_Nr
- DTR_Gb_Nr
- InD_Version
- InD_Offline_Id
- InD_Stream_Id
- InD_Datarec_Nr
- InD_Gb_Nr

Unique constraint:

- DTR_Filename
- InD_Filename

Tables and views used:

- logger.datarec_datarec
- logger.datarec_logger
- *logger.run_data*
    - logger.run_logger
    - descript.run_type_descript
    - kloemaps.torta_map
- descript.stream_offline

logger.dtr_base_raws_data

This view contains the relations between datarec files and the base raw files they were produced from; if a datarec file was produced from other datarec files, the base raw files are found following the path down to the leaves. It also contains information about the resulting datarec files, information about the base raw files and information about the runs they belong to.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Nr	integer	no	Run number
Run_Type_Id	integer	no	Run type identifier; numeric and human readable
Run_Type_Code	varchar(8)		
WaitSync	integer (as boolean)	no	Need the DAQ farm builder wait for the sync cycle before validating the arrived data?
Run_Start_Time	timestamp	no	Run active period Run_End_Time can be used to check if a run is still active or not (NULL value).
Run_End_Time	timestamp	yes	
Zero_Sup	integer (as boolean)	no	Was zero suppression in use?
Run_On_Disk	integer (as boolean)	no	Any physics data produced during the run?
Run_Events	integer	yes	Number of physics events acquired during the run
Torta_Name	varchar(16)	yes	Name of the TORTA configuration used during the run
Run_Normal_End	integer (as boolean)	yes	Run ended on error? (*not of the value*)
DAFNE_Im	real	yes	DAΦNE beam currents (in mA) at start of run (Im – e^-, Ip – e^+)
DAFNE_Ip			
DAFNE_LTm	integer	yes	DAΦNE beam lifetimes (in secs) at start of run (LTm – e^-, LTp – e^+)
DAFNE_LTp			
DAFNE_TSW1	integer (32 bit)	yes	Bit pattern of the DAΦNE beam (bunch crossing) Use as a 128 bit value
DAFNE_TSW2			
DAFNE_TSW3			
DAFNE_TSW4			
DAFNE_Fill	integer	yes	DAΦNE fill number at start of run
DAFNE_Vacuum	real	yes	Vacuum in the KLOE interaction region at start of run
Magnet_Current	real	yes	Current (in A) in the KLOE magnet solenoid (at start of run)

Field name	*Field type*	*Can be NULL?*	*Description*
L3_BhaBha	integer	yes	BhaBhas found during the run by the L3 filters and by the Trigger Monitors
TrgMon_BhaBha			
L3_Luminosity	real	yes	Average luminosity (in $cm^{-2}s^{-1}$) of the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Luminosity			
L3_Int_Lum	real	yes	Luminosity (in cm^{-2} or $10^{-33}nb^{-1}$) integrated during the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Int_Lum			
Calib_Word	integer (32 bit)	no	Bit pattern indicating the current status of the calibration for the run
Reconstruct_Id	integer	yes	Vote given by a human operator
Run_Comment	varchar(40)	yes	Comment given by a human operator
DTR_Version	integer	no	Version of the reconstruction executable
DTR_Offline_Id	char(3)	no	Reconstruction machine identifier
DTR_Datarec_Nr	integer	no	Reconstruction process identifier
DTR_Stream_Id	integer	no	Datarec file stream identifier; numeric and human readable
DTR_Stream_Code	varchar(8)		
DTR_Gb_Nr	integer	no	Datarec file sequential number
DTR_Filename	varchar(64)	no	Datarec file name
DTR_Creation_Time	timestamp	no	Datarec file creation time
DTR_Closure_Time	timestamp	yes	Datarec file closure time Can be used to find out if the file is still being written (NULL value)
DTR_First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the reconstruction process.
DTR_Last			
DTR_Size	integer	yes	Datarec file size
DTR_Total_Events	integer	yes	Number of events inside the datarec file
DTR_Archived	integer	no	Is the datarec file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
DTR_TServer_id	char(3)	yes	Filespace where the datarec file is archived.
DTR_Filespace_Id	integer		
DTR_Analyzed	integer	yes	Not used at the moment.
Raw_Farm_Id	char(3)	no	Acquisition machine identifier
Raw_Receiver_Nr	integer	no	Builder identifier
Raw_Builder_Nr			
Raw_Stream_Id	integer	no	Raw stream identifier; numeric and human readable
Raw_Stream_Code	varchar(8)		

Field name	*Field type*	*Can be NULL?*	*Description*
Raw_Stream_Analyze	integer (as boolean)	no	If false, the raw file should not have been reconstructed.
Raw_Stream_AllData	integer (as boolean)	no	If false, the raw file could have been produced using only a subset of the available events (i.e. by a spy process).
Raw_Recorder_Nr	integer	no	Recorder number
Raw_Gb_Nr	integer	no	Raw file sequential number
Raw_Filename	varchar(64)	no	Raw file name
Raw_Creation_Time	timestamp	no	Raw file creation time
Raw_Closure_Time	timestamp	yes	Raw file closure time Can be used to find out if the file is still being written (NULL value)
Raw_First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the recorder.
Raw_Last			
Raw_Size	integer	yes	Raw file size
Raw_Total_Events	integer	yes	Number of events inside the raw file
Raw_Archived	integer	no	Is the raw file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
Raw_TServer_id	char(3)	yes	Filespace where the raw file is archived.
Raw_Filespace_Id	integer		
Raw_Analyzed	integer	yes	The highest version of datarec used to reconstruct the raw file (can be NULL if the raw file was never fully reconstructed)

Primary key:

- Run_Nr
- DTR_Version
- DTR_Offline_Id
- DTR_Stream_Id
- DTR_Datarec_Nr
- DTR_Gb_Nr
- Raw_Farm_Id
- Raw_Receiver_Nr
- Raw_Builder_Nr
- Raw_Stream_Id
- Raw_Recorder_Nr
- Raw_Gb_Nr

Unique constraint:

- DTR_Filename
- Raw_Filename

Tables and views used:

- logger.datarec_base_raws
- logger.datarec_logger
- logger.raw_logger
- *logger.run_data*
  - logger.run_logger
  - descript.run_type_descript
  - kloemaps.torta_map
- descript.stream_descript
- descript.stream_offline

logger.mc_runs_data

This view contains information about the way mcraw files are produced by the Monte Carlo generators.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
MCCard_Id	integer	no	Run card identifier; numeric and human readable
MCCard_Code	varchar(16)		
MCCard_Grp_Id	integer	no	Run card group identifier; numeric and human readable
MCCard_Grp_Code	varchar(60)		
MCRun_Nr	integer	no	Run number relative to the MC card
Seed_Stream	integer	no	Identifier of the seed stream relative to the MC card
End_Seed1	integer (32 bit)	yes[a]	Seed to be used by the next generation on the same seed stream
End_Seed2			
Start_Time	timestamp	no	Generation start time
End_Time	timestamp	yes	Generation end time Can be used to find out if generation is still in progress (NULL value)
MC_Version	integer	no	Version of the Monte Carlo generator executable
Offline_Id	char(3)	no	Generator machine identifier
Total_Events	integer	yes	Number of events produced
Nr_Trigs	integer	no	Number of triggers issued by the MC generator

Primary key:

- MCCard_Id
- MCRun_Nr

Tables used:

- logger.mc_runs
- descript.mccard_base
- descript.mccard_group

---

[a] can be NULL only if generation is still in progress

logger.mc_runstreams_data

This view contains the number of events produced for every physics stream for every mcraw file and also the information regarding the way the files were produced.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
MCCard_Id	integer	no	Run card identifier; numeric and human readable
MCCard_Code	varchar(16)		
MCCard_Grp_Id	integer	no	Run card group identifier; numeric and human readable
MCCard_Grp_Code	varchar(60)		
MCRun_Nr	integer	no	Run number relative to the MC card
MCStream_Id	integer	no	Physics stream identifier; numeric and human readable
MCStream_Code	varchar(15)		
MCStream_Grp_Id	integer	no	Physics stream identifier; numeric and human readable
MCStream_Grp_Code	varchar(15)		
Seed_Stream	integer	no	Identifier of the seed stream relative to the MC card
End_Seed1	integer (32 bit)	no	Seed to be used by the next generation on the same seed stream
End_Seed2			
Start_Time	timestamp	no	Generation start time
End_Time	timestamp	no	Generation end time
MC_Version	integer	no	Version of the Monte Carlo generator executable
Offline_Id	char(3)	no	Generator machine identifier
MCStream_Events	integer	no	How many events are part of the physics stream
Total_Events	integer	no	Number of events produced
Nr_Trigs	integer	no	Number of triggers issued by the MC generator

Primary key:

- MCCard_Id
- MCRun_Nr
- MCStream_Id

Tables and views used:

- logger.mc_streams
- *logger.mc_runs_data*
    - logger.mc_runs
    - descript.mccard_base
    - descript.mccard_group
- descript.mcstream_descript
- descript.mcstream_group

logger.mc_data

This view contains the list of all the mc files produced including the way the intermediate mcraw files were generated by the MC generators.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
MCCard_Id	integer	no	Run card identifier; numeric and human readable
MCCard_Code	varchar(16)		
MCCard_Grp_Id	integer	no	Run card group identifier; numeric and human readable
MCCard_Grp_Code	varchar(60)		
MCRun_Nr	integer	no	Run number relative to the MC card
Seed_Stream	integer	no	Identifier of the seed stream relative to the MC card
End_Seed1	integer (32 bit)	no	Seed to be used by the next generation on the same seed stream
End_Seed2			
MC_Start_Time	timestamp	no	Generation start time
MC_End_Time	timestamp	no	Generation end time
MC_Version	integer	no	Version of the Monte Carlo generator executable
MC_Offline_Id	char(3)	no	Generator machine identifier
Datarec_Version	integer	no	Version of the reconstruction executable
Datarec_Offline_Id	char(3)	no	Reconstruction machine identifier
Filename	varchar(64)	no	File name
Total_Events	integer	no	Number of events produced
Nr_Trigs	integer	no	Number of triggers issued by the MC generator
Size	integer	yes	File size
Creation_Time	timestamp	no	Creation time
Closure_Time	timestamp	yes	Closure time Can be used to find out if the file is still being written (NULL value)
Archived	integer	no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_Id	char(3)	yes	Filespace where the file is archived.
Filespace_Id	integer		
Analyzed	integer	yes	Not used at the moment.
Nr_recalled	integer	no	Number of copies on user recall areas

Primary key:

- MCCard_Id
- MCRun_Nr

Unique constraint:

- Filename

Tables and view used:

- logger.mc_logger
- *logger.mc_runs_data*
    - logger.mc_runs
    - descript.mccard_base
    - descript.mccard_group

logger.mc_streams_data

This view contains the number of events produced for every physics stream for every mc file and also all the information regarding the files themselves, including how the intermediate mcraw files were generated by the MC generators.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
MCCard_Id	integer	no	Run card identifier; numeric and human readable
MCCard_Code	varchar(16)		
MCCard_Grp_Id	integer	no	Run card group identifier; numeric and human readable
MCCard_Grp_Code	varchar(60)		
MCRun_Nr	integer	no	Run number relative to the MC card
MCStream_Id	integer	no	Physics stream identifier; numeric and human readable
MCStream_Code	varchar(15)		
MCStream_Grp_Id	integer	no	Physics stream identifier; numeric and human readable
MCStream_Grp_Code	varchar(15)		
Seed_Stream	integer	no	Identifier of the seed stream relative to the MC card
End_Seed1	integer (32 bit)	no	Seed to be used by the next generation on the same seed stream
End_Seed2			
MC_Start_Time	timestamp	no	Generation start time
MC_End_Time	timestamp	no	Generation end time
MC_Version	integer	no	Version of the Monte Carlo generator executable
MC_Offline_Id	char(3)	no	Generator machine identifier
Datarec_Version	integer	no	Version of the reconstruction executable
Datarec_Offline_Id	char(3)	no	Reconstruction machine identifier
Filename	varchar(64)	no	File name
MCStream_Events	integer	no	How many events are part of the physics stream
Total_Events	integer	no	Number of events produced
Nr_Trigs	integer	no	Number of triggers issued by the MC generator
Size	integer	yes	File size
Creation_Time	timestamp	no	Creation time
Closure_Time	timestamp	yes	Closure time Can be used to find out if the file is still being written (NULL value)

Field name	***Field type***	***Can be NULL?***	***Description***
Archived	integer	no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_Id	char(3)	yes	Filespace where the file is archived.
Filespace_Id	integer		
Analyzed	integer	yes	Not used at the moment.

Primary key:

- MCCard_Id
- MCRun_Nr
- MCStream_Id

Unique constraint:

- Filename
- MCStream_Id

Tables and views used:

- logger.mc_streams
- *logger.mc_data*
  - logger.mc_logger
  - *logger.mc_runs_data*
    - logger.mc_runs
    - descript.mccard_base
    - descript.mccard_group
- descript.mcstream_descript
- descript.mcstream_group

descript.mccard_descript

This view contains the MC data cards used for the production of new mcraw files. Only the last instance of each data card is present.

Fields description:

Field name	*Field type*	*Can be NULL?*	*Description*
MCCard_Id	integer	no	MC card identifier
Description	varchar(60)	no	MC card description, used for human readable interaction
MCCard_Group	integer	no	MC group identifier
Nr_Trigs	integer	no	Number of triggers to generate at each run
Card	varchar(40k)	no	The data card
Creation_Time	timestamp	no	Creation time

Primary key:

- MCCard_Id

Unique constraint:

- Description

Tables used:

- descript.mccard_base
- descript.mccard_card

Position related views

The views described in this section treats data about the position of files on the disk areas. Since there are four different kinds of files; raw, datarec, mcraw and mc, and 2 different types of disks; local and recall disks, a two dimensional classification of the available views comes natural:

	raw	***datarec***	***mcraw***	***mc***
local disk pool	status.raw_online_data	status.dtr_offline_data	status.mc_temp_data	status.mc_offline_data
recall disk pool	status.raw_recalled_data	status.dtr_recalled_data		status.mc_recalled_data

However, since this is also a 2D paper, a linear classification is needed in order to make a total order of the views. The classification of choice is the one based on the kind of files.

status.raw_online_data

This view contains the list of raw files that are currently residing on the DAQ online disk pool. It contains both the data about the files themselves and the data about the runs they belong to.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Nr	integer	no	Run number
Farm_Id	char(3)	no	Acquisition machine identifier
Receiver_Nr	integer	no	Builder identifier
Builder_Nr			
Stream_Id	integer	no	Stream identifier; numeric and human readable
Stream_Code	varchar(8)		
Stream_Analyze	integer (as boolean)	no	If false, the file will not be reconstructed.
Stream_AllData	integer (as boolean)	no	If false, the file could have been produced using only a subset of the available events (i.e. by a spy process).
Recorder_Nr	integer	no	Recorder number
Gb_Nr	integer	no	Sequential number
Filename	varchar(64)	no	File name
Run_Type_Id	integer	no	Run type identifier; numeric and human readable
Run_Type_Code	varchar(8)		
WaitSync	integer (as boolean)	no	Need the DAQ farm builder wait for the sync cycle before validating the arrived data?
Run_Start_Time	timestamp	no	Run active period Run_End_Time can be used to check if a run is still active or not (NULL value).
Run_End_Time	timestamp	yes	
Zero_Sup	integer (as boolean)	no	Was zero suppression in use?
Run_On_Disk	integer (as boolean)	no	Any physics data produced during the run?
Run_Events	integer	yes	Number of physics events acquired during the run
Torta_Name	varchar(16)	yes	Name of the TORTA configuration used during the run
Run_Normal_End	integer (as boolean)	yes	Run ended on error? (*not of the value*)
DAFNE_Im	real	yes	DAΦNE beam currents (in mA) at start of run (Im – e^-, Ip – e^+)
DAFNE_Ip			

Field name	*Field type*	*Can be NULL?*	*Description*
DAFNE_LTm	integer	yes	DAΦNE beam lifetimes (in secs) at start of run (LTm – e^-, LTp – e^+)
DAFNE_LTp			
DAFNE_TSW1	integer (32 bit)	yes	Bit pattern of the DAΦNE beam (bunch crossing) Use as a 128 bit value
DAFNE_TSW2			
DAFNE_TSW3			
DAFNE_TSW4			
DAFNE_Fill	integer	yes	DAΦNE fill number at start of run
DAFNE_Vacuum	real	yes	Vacuum in the KLOE interaction region at start of run
Magnet_Current	real	yes	Current (in A) in the KLOE magnet solenoid (at start of run)
L3_BhaBha	integer	yes	BhaBhas found during the run by the L3 filters and by the Trigger Monitors
TrgMon_BhaBha			
L3_Luminosity	real	yes	Average luminosity (in cm^{-2}s^{-1}) of the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Luminosity			
L3_Int_Lum	real	yes	Luminosity (in cm^{-2} or 10^{-33}nb^{-1}) integrated during the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Int_Lum			
Calib_Word	integer (32 bit)	no	Bit pattern indicating the current status of the calibration for the run
Reconstruct_Id	integer	yes	Vote given by a human operator
Run_Comment	varchar(40)	yes	Comment given by a human operator
Creation_Time	timestamp	no	Creation time
Closure_Time	timestamp	yes	Closure time Can be used to find out if the file is still being written (NULL value)
First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the recorder.
Last			
Size	integer	yes	File size
Total_Events	integer	yes	Number of events inside the file
Archived	integer	no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_id	char(3)	yes	Filespace where the file is archived.
Filespace_Id	integer		
Analyzed	integer	yes	The highest version of datarec used to reconstruct this file (NULL if never reconstructed)
FarmDir_Id	integer	no	Disk pool area identifier

Field name	***Field type***	***Can be NULL?***	***Description***
Nr_Act_Acc	integer	no	Number of processes currently accessing this copy of the file

Primary key:

- Run_Nr
- Farm_Id
- Receiver_Nr
- Builder_Nr
- Stream_Id
- Recorder_Nr
- Gb_Nr

Unique constraint:

- Filename

Tables and views used:

- status.raw_online
- *logger.raw_data*
  - logger.raw_logger
  - *logger.run_data*
    - logger.run_logger
    - descript.run_type_descript
    - kloemaps.torta_map
  - descript.stream_descript

status.raw_recalled_data

This view contains the list of all the copies of raw files that are currently residing on the recall disk areas. It contains both the data about the files themselves and the data about the runs they belong to.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Nr	integer	no	Run number
Farm_Id	char(3)	no	Acquisition machine identifier
Receiver_Nr	integer	no	Builder identifier
Builder_Nr			
Stream_Id	integer	no	Stream identifier; numeric and human readable
Stream_Code	varchar(8)		
Stream_Analyze	integer (as boolean)	no	If false, the file will not be reconstructed.
Stream_AllData	integer (as boolean)	no	If false, the file could have been produced using only a subset of the available events (i.e. by a spy process).
Recorder_Nr	integer	no	Recorder number
Gb_Nr	integer	no	Sequential number
Filename	varchar(64)	no	File name
Run_Type_Id	integer	no	Run type identifier; numeric and human readable
Run_Type_Code	varchar(8)		
WaitSync	integer (as boolean)	no	Need the DAQ farm builder wait for the sync cycle before validating the arrived data?
Run_Start_Time	timestamp	no	Run active period Run_End_Time can be used to check if a run is still active or not (NULL value).
Run_End_Time	timestamp	yes	
Zero_Sup	integer (as boolean)	no	Was zero suppression in use?
Run_On_Disk	integer (as boolean)	no	Any physics data produced during the run?
Run_Events	integer	yes	Number of physics events acquired during the run
Torta_Name	varchar(16)	yes	Name of the TORTA configuration used during the run
Run_Normal_End	integer (as boolean)	yes	Run ended on error? (*not of the value*)
DAFNE_Im	real	yes	DAΦNE beam currents (in mA) at start of run (Im – e^-, Ip – e^+)
DAFNE_Ip			

Field name	***Field type***	***Can be NULL?***	***Description***
DAFNE_LTm	integer	yes	DAΦNE beam lifetimes (in secs) at start of run (LTm – e^-, LTp – e^+)
DAFNE_LTp			
DAFNE_TSW1	integer (32 bit)	yes	Bit pattern of the DAΦNE beam (bunch crossing) Use as a 128 bit value
DAFNE_TSW2			
DAFNE_TSW3			
DAFNE_TSW4			
DAFNE_Fill	integer	yes	DAΦNE fill number at start of run
DAFNE_Vacuum	real	yes	Vacuum in the KLOE interaction region at start of run
Magnet_Current	real	yes	Current (in A) in the KLOE magnet solenoid (at start of run)
L3_BhaBha	integer	yes	BhaBhas found during the run by the L3 filters and by the Trigger Monitors
TrgMon_BhaBha			
L3_Luminosity	real	yes	Average luminosity (in $cm^{-2}s^{-1}$) of the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Luminosity			
L3_Int_Lum	real	yes	Luminosity (in cm^{-2} or $10^{-33}nb^{-1}$) integrated during the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Int_Lum			
Calib_Word	integer (32 bit)	no	Bit pattern indicating the current status of the calibration for the run
Reconstruct_Id	integer	yes	Vote given by a human operator
Run_Comment	varchar(40)	yes	Comment given by a human operator
Creation_Time	timestamp	no	Creation time
Closure_Time	timestamp	yes	Closure time Can be used to find out if the file is still being written (NULL value)
First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the recorder.
Last			
Size	integer	yes	File size
Total_Events	integer	yes	Number of events inside the file
Archived	integer	no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_id	char(3)	yes	Filespace where the file is archived.
Filespace_Id	integer		
Analyzed	integer	yes	The highest version of datarec used to reconstruct this file (NULL if never reconstructed)

Field name	***Field type***	***Can be NULL?***	***Description***
DServer_Id	char(3)	no	Disk pool area identifier
DSrvDir_Id	integer		
Nr_Act_Acc	integer	no	Number of processes currently accessing this copy of the file
Recalled_Time	timestamp	yes	Last access time If NULL, the file is not yet available on disk

Primary key:

- Run_Nr
- Farm_Id
- Receiver_Nr
- Builder_Nr
- Stream_Id
- Recorder_Nr
- Gb_Nr
- DServer_Id
- DSrvDir_Id

Unique constraint:

- Filename
- DServer_Id
- DSrvDir_Id

Tables and views used:

- status.raw_recalled
- *logger.raw_data*
  - logger.raw_logger
  - *logger.run_data*
    - logger.run_logger
    - descript.run_type_descript
    - kloemaps.torta_map
  - descript.stream_descript

status.dtr_offline_data

This view contains the list of datarec files currently residing on the offline disk pool. It contains both the data about the files themselves and the data about the runs they belong to.

Fields description:

Field name	*Field type*	*Can be NULL?*	*Description*
Run_Nr	integer	no	Run number
Version	integer	no	Version of the reconstruction executable
Offline_Id	char(3)	no	Reconstruction machine identifier
Datarec_Nr	integer	no	Process identifier
Stream_Id	integer	no	Stream identifier; numeric and human readable
Stream_Code	varchar(8)		
Gb_Nr	integer	no	Sequential number
Filename	varchar(64)	no	File name
Run_Type_Id	integer	no	Run type identifier; numeric and human readable
Run_Type_Code	varchar(8)		
WaitSync	integer (as boolean)	no	Need the DAQ farm builder wait for the sync cycle before validating the arrived data?
Run_Start_Time	timestamp	no	Run active period Run_End_Time can be used to check if a run is still active or not (NULL value).
Run_End_Time	timestamp	yes	
Zero_Sup	integer (as boolean)	no	Was zero suppression in use?
Run_On_Disk	integer (as boolean)	no	Any physics data produced during the run?
Run_Events	integer	yes	Number of physics events acquired during the run
Torta_Name	varchar(16)	yes	Name of the TORTA configuration used during the run
Run_Normal_End	integer (as boolean)	yes	Run ended on error? (*not of the value*)
DAFNE_Im	real	yes	DAΦNE beam currents (in mA) at start of run (Im – e^-, Ip – e^+)
DAFNE_Ip			
DAFNE_LTm	integer	yes	DAΦNE beam lifetimes (in secs) at start of run (LTm – e^-, LTp – e^+)
DAFNE_LTp			
DAFNE_TSW1	integer (32 bit)	yes	Bit pattern of the DAΦNE beam (bunch crossing) Use as a 128 bit value
DAFNE_TSW2			
DAFNE_TSW3			
DAFNE_TSW4			

Field name	*Field type*	*Can be NULL?*	*Description*
DAFNE_Fill	integer	yes	DAΦNE fill number at start of run
DAFNE_Vacuum	real	yes	Vacuum in the KLOE interaction region at start of run
Magnet_Current	real	yes	Current (in A) in the KLOE magnet solenoid (at start of run)
L3_BhaBha	integer	yes	BhaBhas found during the run by the L3 filters and by the Trigger Monitors
TrgMon_BhaBha			
L3_Luminosity	real	yes	Average luminosity (in $cm^{-2}s^{-1}$) of the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Luminosity			
L3_Int_Lum	real	yes	Luminosity (in cm^{-2} or $10^{-33}nb^{-1}$) integrated during the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Int_Lum			
Calib_Word	integer (32 bit)	no	Bit pattern indicating the current status of the calibration for the run
Reconstruct_Id	integer	yes	Vote given by a human operator
Run_Comment	varchar(40)	yes	Comment given by a human operator
Creation_Time	timestamp	no	Creation time
Closure_Time	timestamp	yes	Closure time Can be used to find out if the file is still being written (NULL value)
First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the process.
Last			
Size	integer	yes	File size
Total_Events	integer	yes	Number of events inside the file
Archived	integer	no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_id	char(3)	yes	Filespace where the file is archived.
Filespace_Id	integer		
Analyzed	integer	yes	Not used at the moment.
OffDir_Id	integer	no	Disk pool area identifier
OffSrv_Id	char(3)	no	Offline disk server disk pool area identifier
OffSrvDir_Id	integer		
Nr_Act_Acc	integer	no	Number of processes currently accessing this copy of the file

Primary key:

- Run_Nr
- Version
- Offline_Id
- Datarec_Nr
- Stream_Id
- Gb_Nr

Unique constraint:

- Filename

Tables and views used:

- status.datarec_offline
- descript.offdir_descript
- *logger.datarec_data*
    - logger.datarec_logger
    - *logger.run_data*
        - logger.run_logger
        - descript.run_type_descript
        - kloemaps.torta_map
    - descript.stream_offline

status.dtr_recalled_data

This view contains the list of all the copies of datarec files currently residing on the recall disk areas. It contains both the data about the files themselves and the data about the runs they belong to.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
Run_Nr	integer	no	Run number
Version	integer	no	Version of the reconstruction executable
Offline_Id	char(3)	no	Reconstruction machine identifier
Datarec_Nr	integer	no	Process identifier
Stream_Id	integer	no	Stream identifier; numeric and human readable
Stream_Code	varchar(8)		
Gb_Nr	integer	no	Sequential number
Filename	varchar(64)	no	File name
Run_Type_Id	integer	no	Run type identifier; numeric and human readable
Run_Type_Code	varchar(8)		
WaitSync	integer (as boolean)	no	Need the DAQ farm builder wait for the sync cycle before validating the arrived data?
Run_Start_Time	timestamp	no	Run active period Run_End_Time can be used to check if a run is still active or not (NULL value).
Run_End_Time	timestamp	yes	
Zero_Sup	integer (as boolean)	no	Was zero suppression in use?
Run_On_Disk	integer (as boolean)	no	Any physics data produced during the run?
Run_Events	integer	yes	Number of physics events acquired during the run
Torta_Name	varchar(16)	yes	Name of the TORTA configuration used during the run
Run_Normal_End	integer (as boolean)	yes	Run ended on error? (*not of the value*)
DAFNE_Im	real	yes	DAΦNE beam currents (in mA) at start of run (Im – e^-, Ip – e^+)
DAFNE_Ip			
DAFNE_LTm	integer	yes	DAΦNE beam lifetimes (in secs) at start of run (LTm – e^-, LTp – e^+)
DAFNE_LTp			

Field name	*Field type*	*Can be NULL?*	*Description*
DAFNE_TSW1	integer (32 bit)	yes	Bit pattern of the DAΦNE beam (bunch crossing) Use as a 128 bit value
DAFNE_TSW2			
DAFNE_TSW3			
DAFNE_TSW4			
DAFNE_Fill	integer	yes	DAΦNE fill number at start of run
DAFNE_Vacuum	real	yes	Vacuum in the KLOE interaction region at start of run
Magnet_Current	real	yes	Current (in A) in the KLOE magnet solenoid (at start of run)
L3_BhaBha	integer	yes	BhaBhas found during the run by the L3 filters and by the Trigger Monitors
TrgMon_BhaBha			
L3_Luminosity	real	yes	Average luminosity (in $cm^{-2}s^{-1}$) of the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Luminosity			
L3_Int_Lum	real	yes	Luminosity (in cm^{-2} or $10^{-33}nb^{-1}$) integrated during the run, estimated by the L3 filters and the Trigger Monitors
TrgMon_Int_Lum			
Calib_Word	integer (32 bit)	no	Bit pattern indicating the current status of the calibration for the run
Reconstruct_Id	integer	yes	Vote given by a human operator
Run_Comment	varchar(40)	yes	Comment given by a human operator
Creation_Time	timestamp	no	Creation time
Closure_Time	timestamp	yes	Closure time Can be used to find out if the file is still being written (NULL value)
First	integer (as boolean)	no	Specifies if this is the first/last file in the sequence produced by the process.
Last			
Size	integer	yes	File size
Total_Events	integer	yes	Number of events inside the file
Archived	integer	no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_id	char(3)	yes	Filespace where the file is archived.
Filespace_Id	integer		
Analyzed	integer	yes	Not used at the moment.
DServer_Id	char(3)	no	Disk pool area identifier
DSrvDir_Id	integer		
Nr_Act_Acc	integer	no	Number of processes currently accessing this copy of the file

Field name	***Field type***	***Can be NULL?***	***Description***
Recalled_Time	timestamp	yes	Last access time If NULL, the file is not yet available on disk

Primary key:

- Run_Nr
- Version
- Offline_Id
- Datarec_Nr
- Stream_Id
- Gb_Nr
- DServer_Id
- DSrvDir_Id

Unique constraint:

- Filename
- DServer_Id
- DSrvDir_Id

Tables and views used:

- status.datarec_recalled
- *logger.datarec_data*
  - logger.datarec_logger
  - *logger.run_data*
    - logger.run_logger
    - descript.run_type_descript
    - kloemaps.torta_map
  - descript.stream_offline

status.mc_temp_data

This view contains the list of mcraw files and their position on local disks. It contains both the data about the files themselves and the data about the way they were produced by the Monte Carlo generators.

Fields description:

Field name	*Field type*	*Can be NULL?*	*Description*
MCCard_Id	integer	no	Run card identifier; numeric and human readable
MCCard_Code	varchar(16)		
MCCard_Grp_Id	integer	no	Run card group identifier; numeric and human readable
MCCard_Grp_Code	varchar(60)		
MCRun_Nr	integer	no	Run number relative to the MC card
Seed_Stream	integer	no	Identifier of the seed stream relative to the MC card
End_Seed1	integer (32 bit)	yes[a]	Seed to be used by the next generation on the same seed stream
End_Seed2			
Start_Time	timestamp	no	Generation start time
End_Time	timestamp	yes	Generation end time Can be used to find out if generation is still in progress (NULL value)
MC_Version	integer	no	Version of the Monte Carlo generator executable
Offline_Id	char(3)	no	Generator machine identifier
Total_Events	integer	yes	Number of events produced
Nr_Trigs	integer	no	Number of triggers issued by the MC generator
FilePath	varchar(240)	no	File path (directory + file name)

Primary key:

- MCCard_Id
- MCRun_Nr

Unique constraint:

- Offline_Id
- FilePath

---

[a] can be NULL only if generation is still in progress

Tables used:

- status.mc_temp
- *logger.mc_runs_data*
  - logger.mc_runs
  - descript.mccard_base
  - descript.mccard_group

status.mc_offline_data

This view contains the list of mc files that are currently residing on the offline disk pool. It contains both the data about the files themselves and the data about the way the intermediate mcraw files were produced by the Monte Carlo generators.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
MCCard_Id	integer	no	Run card identifier; numeric and human readable
MCCard_Code	varchar(16)		
MCCard_Grp_Id	integer	no	Run card group identifier; numeric and human readable
MCCard_Grp_Code	varchar(60)		
MCRun_Nr	integer	no	Run number relative to the MC card
Seed_Stream	integer	no	Identifier of the seed stream relative to the MC card
End_Seed1	integer (32 bit)	no	Seed to be used by the next generation on the same seed stream
End_Seed2			
MC_Start_Time	timestamp	no	Generation start time
MC_End_Time	timestamp	no	Generation end time
MC_Version	integer	no	Version of the Monte Carlo generator executable
MC_Offline_Id	char(3)	no	Generator machine identifier
Datarec_Version	integer	no	Version of the reconstruction executable
Datarec_Offline_Id	char(3)	no	Reconstruction machine identifier
Filename	varchar(64)	no	File name
Total_Events	integer	no	Number of events produced
Nr_Trigs	integer	no	Number of triggers issued by the MC generator
Size	integer	yes	File size
Creation_Time	timestamp	no	Creation time
Closure_Time	timestamp	yes	Closure time Can be used to find out if the file is still being written (NULL value)
Archived	integer	no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_Id	char(3)	yes	Filespace where the file is archived.
Filespace_Id	integer		
Analyzed	integer	yes	Not used at the moment.
Datarec_OffDir_Id	integer	no	Disk pool area identifier
Nr_Act_Acc	integer	no	Number of processes currently accessing this copy of the file

Primary key:

- MCCard_Id
- MCRun_Nr

Unique constraint:

- Filename

Tables and view used:

- status.mc_offline
- *logger.mc_data*
  - logger.mc_logger
  - *logger.mc_runs_data*
    - logger.mc_runs
    - descript.mccard_base
    - descript.mccard_group

status.mc_recalled_data

This view contains the list of all the copies of mc files that are currently residing on the recall disk areas. It contains both the data about the files themselves and the data about the way the intermediate mcraw files were produced by the Monte Carlo generators.

Fields description:

Field name	***Field type***	***Can be NULL?***	***Description***
MCCard_Id	integer	no	Run card identifier; numeric and human readable
MCCard_Code	varchar(16)		
MCCard_Grp_Id	integer	no	Run card group identifier; numeric and human readable
MCCard_Grp_Code	varchar(60)		
MCRun_Nr	integer	no	Run number relative to the MC card
Seed_Stream	integer	no	Identifier of the seed stream relative to the MC card
End_Seed1	integer (32 bit)	no	Seed to be used by the next generation on the same seed stream
End_Seed2			
MC_Start_Time	timestamp	no	Generation start time
MC_End_Time	timestamp	no	Generation end time
MC_Version	integer	no	Version of the Monte Carlo generator executable
MC_Offline_Id	char(3)	no	Generator machine identifier
Datarec_Version	integer	no	Version of the reconstruction executable
Datarec_Offline_Id	char(3)	no	Reconstruction machine identifier
Filename	varchar(64)	no	File name
Total_Events	integer	no	Number of events produced
Nr_Trigs	integer	no	Number of triggers issued by the MC generator
Size	integer	yes	File size
Creation_Time	timestamp	no	Creation time
Closure_Time	timestamp	yes	Closure time Can be used to find out if the file is still being written (NULL value)
Archived	integer	no	Is the file archived or archiving? (0 – not archived, 1 – archiving, 2 – archived)
TServer_Id	char(3)	yes	Filespace where the file is archived.
Filespace_Id	integer		
Analyzed	integer	yes	Not used at the moment.
DServer_Id	char(3)	no	Disk pool area identifier
DSrvDir_Id	integer		

Field name	*Field type*	*Can be NULL?*	*Description*
Nr_Act_Acc	integer	no	Number of processes currently accessing this copy of the file
Recalled_Time	timestamp	yes	Last access time If NULL, the file is not yet available on disk

Primary key:

- MCCard_Id
- MCRun_Nr
- DServer_Id
- DSrvDir_Id

Unique constraint:

- Filename
- DServer_Id
- DSrvDir_Id

Tables and view used:

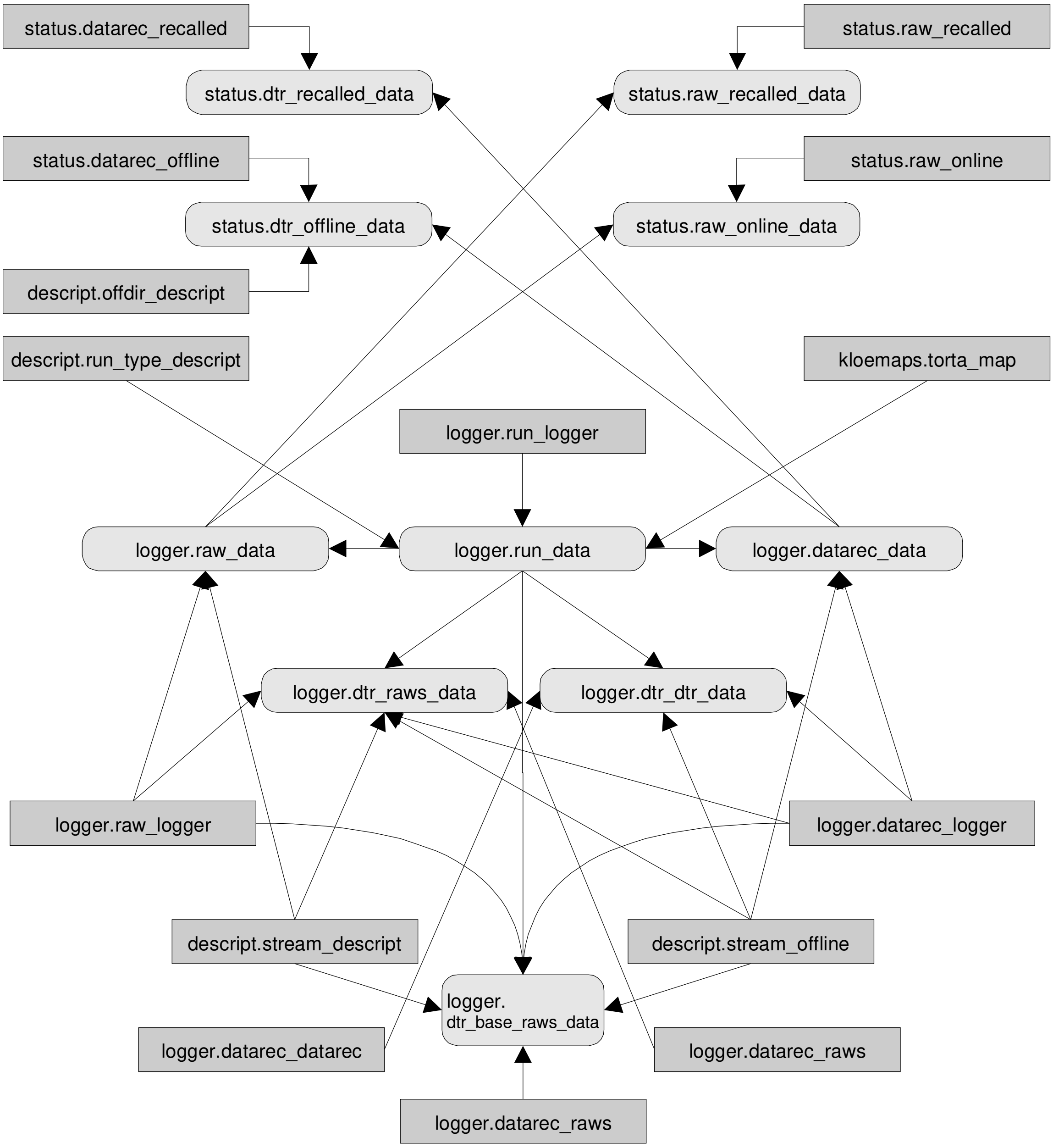
- status.mc_recalled
- *logger.mc_data*
  - logger.mc_logger
  - *logger.mc_runs_data*
    - logger.mc_runs
    - descript.mccard_base
    - descript.mccard_group

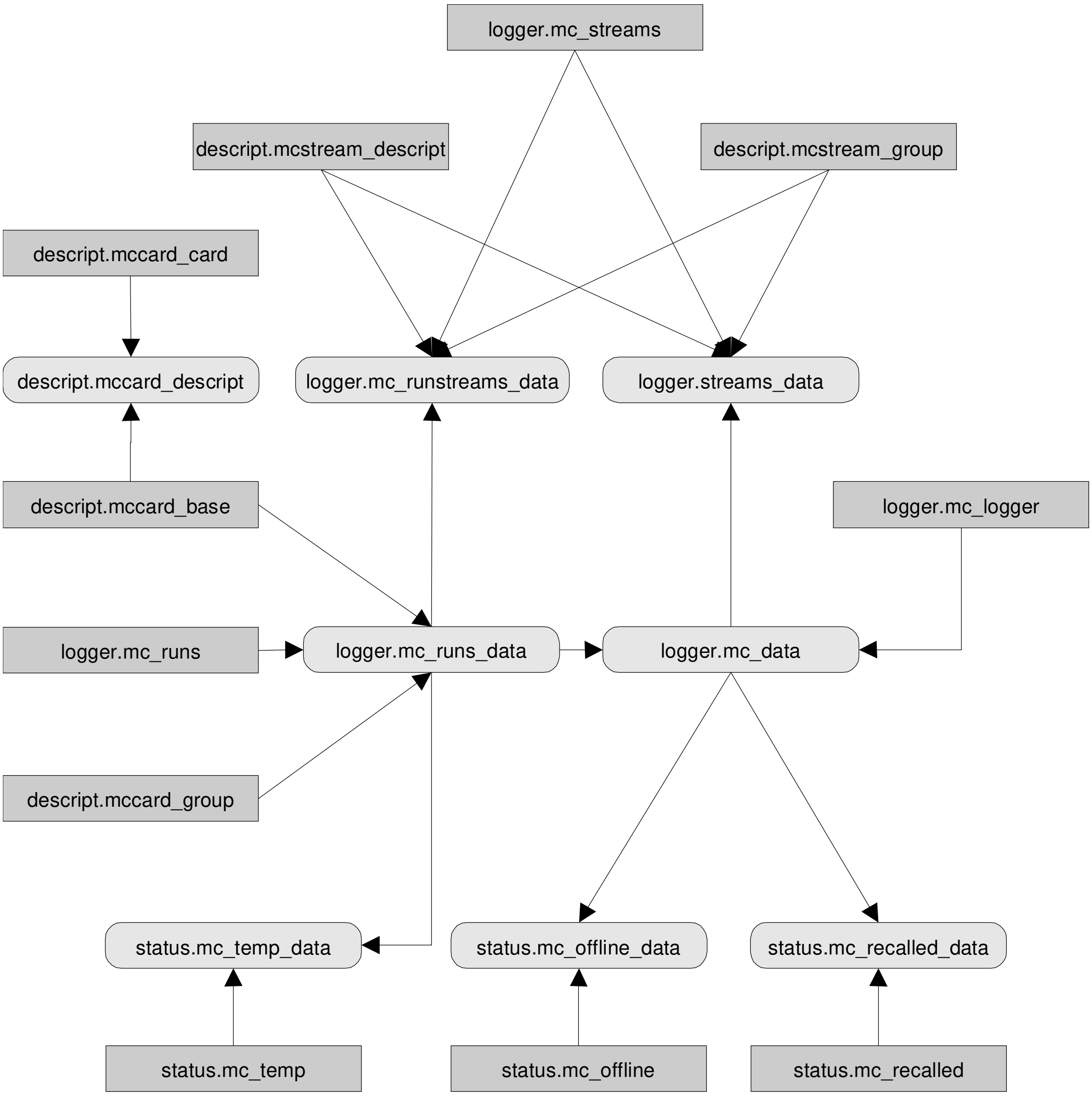
Summary of views

View Name	*Description*
descript.mccard_descript	This view contains the MC data cards used for the production of new mcraw files. Only the last instance of each data card is present.
logger.datarec_data	This view contains the list of all the datarec files produced during the reconstruction process. It contains both the data about the files themselves and the data about the runs they belong to.
logger.dtr_raws_data	This view contains the relations between raw and datarec files of the datarec files that were reconstructed starting from raw files. It contains information about the resulting datarec files, information about the source raw files and information about the runs they belong to.
logger.dtr_dtr_data	This view contains the relations between datarec files of the datarec files that were reconstructed starting from other datarec files. It contains information about both the source and the resulting datarec files and information about the runs they belong to.
logger.dtr_base_raws_data	This view contains the relations between datarec files and the base raw files they were produced from; if a datarec file was produced from other datarec files, the base raw files are found following the path down to the leaves. It also contains information about the resulting datarec files, information about the base raw files and information about the runs they belong to.
logger.mc_data	This view contains the list of all the mc files produced including the way the intermediate mcraw files were generated by the MC generators.
logger.mc_runs_data	This view contains information about the way mcraw files are produced by the Monte Carlo generators.
logger.mc_runstreams_data	This view contains the number of events produced for every physics stream for every mcraw file and also the information regarding the way the files were produced.
logger.mc_streams_data	This view contains the number of events produced for every physics stream for every mc file and also all the information regarding the files themselves, including how the intermediate mcraw files were generated by the MC generators.
logger.raw_data	This view contains the list of all the raw files produced during the data taking. It contains both the data about the files themselves and the data about the runs they belong to.
logger.run_data	This view contains the list of runs taken during the life of the experiment.
status.dtr_offline_data	This view contains the list of datarec files currently residing on the offline disk pool. It contains both the data about the files themselves and the data about the runs they belong to.
status.dtr_recalled_data	This view contains the list of all the copies of datarec files currently residing on the recall disk areas. It contains both the data about the files themselves and the data about the runs they belong to.

View Name	*Description*
status.mc_offline_data	This view contains the list of mc files that are currently residing on the offline disk pool. It contains both the data about the files themselves and the data about the way the intermediate mcraw files were produced by the Monte Carlo generators.
status.mc_recalled_data	This view contains the list of all the copies of mc files that are currently residing on the recall disk areas. It contains both the data about the files themselves and the data about the way the intermediate mcraw files were produced by the Monte Carlo generators.
status.mc_temp_data	This view contains the list of mcraw files and their position on local disks. It contains both the data about the files themselves and the data about the way they were produced by the Monte Carlo generators.
status.raw_online_data	This view contains the list of raw files that are currently residing on the DAQ online disk pool. It contains both the data about the files themselves and the data about the runs they belong to.
status.raw_recalled_data	This view contains the list of all the copies of raw files that are currently residing on the recall disk areas. It contains both the data about the files themselves and the data about the runs they belong to.

Tables used inside the views




logger.mc_streams
descript.mcstream_descript
descript.mcstream_group
descript.mccard_card
descript.mccard_descript
logger.mc_runstreams_data
logger.streams_data
descript.mccard_base
logger.mc_logger
logger.mc_runs
logger.mc_runs_data
logger.mc_data
descript.mccard_group
status.mc_temp_data
status.mc_offline_data
status.mc_recalled_data
status.mc_temp
status.mc_offline
status.mc_recalled

Conclusion

The number of tables and views defined in the KLOE bookkeeping database can frighten a novice user. However the number of tables, or better to say views, needed to solve any specific problem is normally very low; typically a single view contains all the information a generic user may need to solve a task. Nevertheless, all the tables described in this paper are needed to keep track of the data produced in KLOE, but only a few power users will ever need to access all of them.

This paper was thought as a reference manual for the bookkeeping part of the KLOE database. I hope it has accomplished its task. Anyhow, any suggestion and comment aimed at improving this paper will be well accepted.