Heliophysics Integrated Observatory

Project No.: 238969
Call: FP7-INFRA-2008-2

Metadata Server Infrastructure
HEC Architecture

Title:	**Metadata Server Infrastructure – HEC Architecture**
Document No.:	HELIO-INAF-S2-001-SP
Date:	15 March 2010
Editor:	**A. Marassi, A. Santin, M. Messerotti** (INAF-OATS)
Contributors:	**Mauro Messerotti (INAF-OATS)** **Antonio Volpicelli (INAF-OATO)** **Silvio Giordano (INAF-OATO)** **Andrej Santin (INAF-OATS)**
Distribution:	HELIO Project, SOTERIA Project

CAPACITIES

e-infrastructure

Version	**Date**	**Released by**	**Detail**
1.0	15/03/10	M. Messerotti	Architecture of HEC-Release 1

Note: This document will continue to undergo revisions during the implementation phase of HELIO to incorporate changes and improvements.

Contents

List of Figures

List of Tables

This page intentionally left blank.

1 Introduction

The Heliophysics Event Catalogue (HEC) acts as a special provider to HELIO, capable to answer, both at human and at machine level, SQL queries relevant to metadata specific of selected heliophysics data catalogues (see Table 1). Such data catalogues are ingested into a RDBMS and automatically updated on a daily basis from the source provider when non-static ones.

To speed up the HELIO R1 test phase, an Heliophysics Event Catalogue Release 1 (HEC-R1) has been developed upon the EGSO Solar Event Catalogue (SEC), but with an interface compliant with the related proposal in the preliminary HELIO Interface Document. The catalogues managed by HEC-R1 are the same as the SEC catalogue set and since it is a clone of the existing service, it has therefore the same capabilities as SEC.

The human interface (GUI) of the two services is available respectively at:
EGSO/SEC : **http://sec.oats.inaf.it**
HELIO/HEC : **http://hec.oats.inaf.it**

HEC-R2 has been developed in parallel and, according to the requirements, in its pre-release implementation is able to process SQL queries like e.g.:
SELECT * from gev_list WHERE time between T1 and T2 AND xray_class gt "M1"

Until HEC-R2 will become available, a VOTable generated by HEC-R1 resolves the same SQL query for test purposes.

Purpose	Input	Output
Provide a list of events that meet the selection criteria within the requested time interval Note: In principle the HEC could be run on a stand-alone basis and the resulting list imported into the HELIO search engine	*Query based on:* Search time range Type of event list ***Qualifiers dependent on type of list***	VOTable or text file with selected list elements for requested time range. Parameters determines output format, coordinate type, etc.

Table 1 HEC functional requirements

This document describes the architecture of HEC-R1 and illustrates its usage.

1.1 Relevant Documents

- The concepts behind HELIO are described in the "Concepts Document" – https://grid.ie/helio/wiki/FrontPage
- The overall architecture is described in the "Architecture Document" – https://grid.ie/helio/wiki/FrontPage
- Preliminary Interfaces are specified in the "Interface Document" – https://grid.ie/helio/wiki/FrontPage

2 General Architecture

The general architecture of HEC-R1 is reported in Figure 1Figure 1.

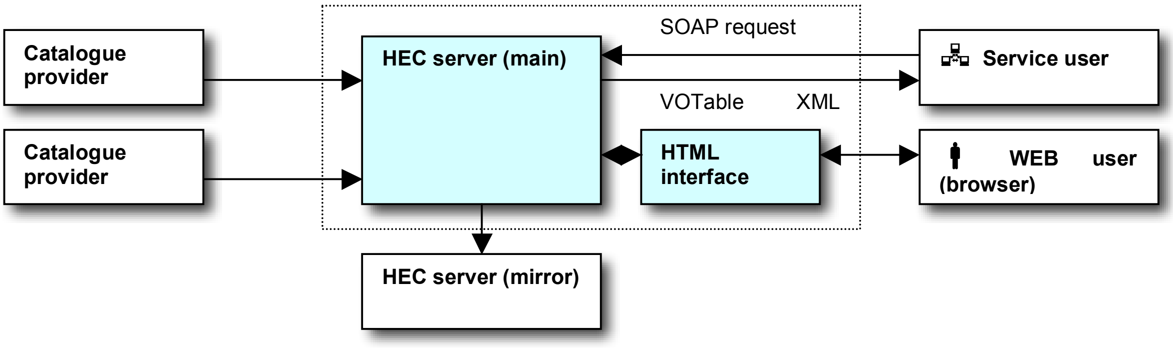


Figure 1 General architecture of HEC-R1

The following operations are relevant to the HEC system:

a) Data are periodically downloaded from catalogue providers, then converted and stored into the HEC database.
b) The HEC server can be mirrored by other HEC service providers.
c) A service user (e.g. a HELIO broker) performs a request (SOAP format) and gets an aswer (VOTable XML).
d) A web user (e.g. people browsing on internet) retrieves data in HTML. The server itself manages the presentation of XML data.

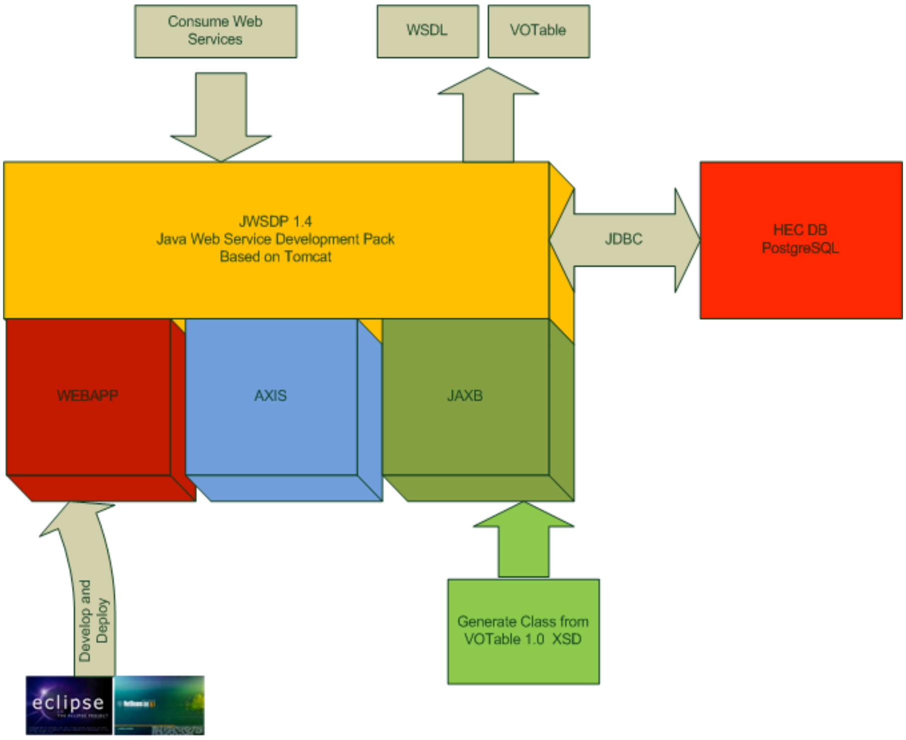


Figure 2 HEC-R1 functional diagram

The functional diagram of the HEC-R1 architecture is reported in Figure 2.

Its features can be summarized as follows:

- The HEC-R1 Web Services architecture is based on the SOAP (Simple Object Access Protocol) XML Web Services technology.
- It uses an AXIS engine to manage the SOAP envelope and generate WSDL.
- JAXB is used to generate the java classes from VOTable.xsd (v. 1.0) schema
  %JAXB_HOME%\bin\xjc.sh VOTable.xsd
- This will generate all VOTable java classes.
- The datasets from PostgreSQL are binded on a VOTable object instance and returned as string in the "sql" web service method detailed in the following.

2.1 References

Some references for the relevant formats and protocols:

- VOTable (Virtual Observatory Table) : http://www.us-vo.org/VOTable/
- XML (eXtensible Markup Language), RDF (Resource Description Framework), SOAP (Simple Object Access Protocol), etc.: http://www.w3.org/TR/

2.2 Current Software Versions

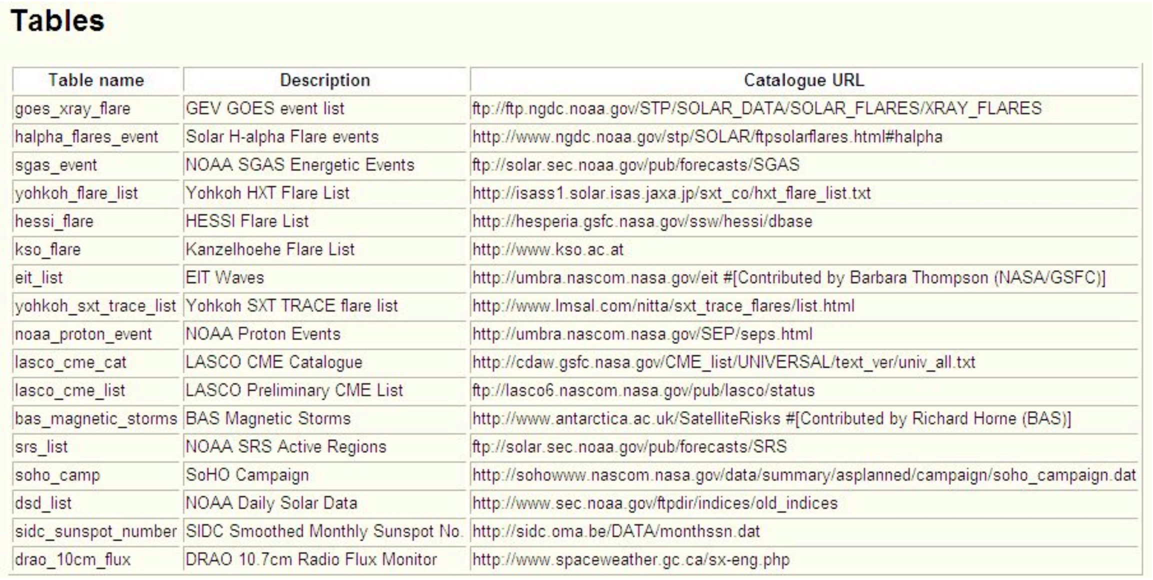
- GNU/Linux Fedora 3
- PHP 4.3.10
- Apache 2.0.52
- PostgreSQL 7.4.7
- NuSOAP 1.67: http://dietrich.ganx4.com/nusoap
  (link to PDF documentation:
  http://dietrich.ganx4.com/download.php?url=/nusoap/downloads/7469_Chap08.pdf)
- JWSDP 1.4 (Java Web Services Development Pack) based on Tomcat
- AXIS 1.2 engine for WS
- JAXB java xml binding [JavaTM Architecture for XML Binding(JAXB) Reference Implementation, (build 1.0.4-b18-fcs)]
- Java: 1.4

3 Database Architecture

Two groups of tables are used:

- Catalogue data tables (one for each data provider);
- Metadata tables describing attributes, catalogue tables and their relationship.

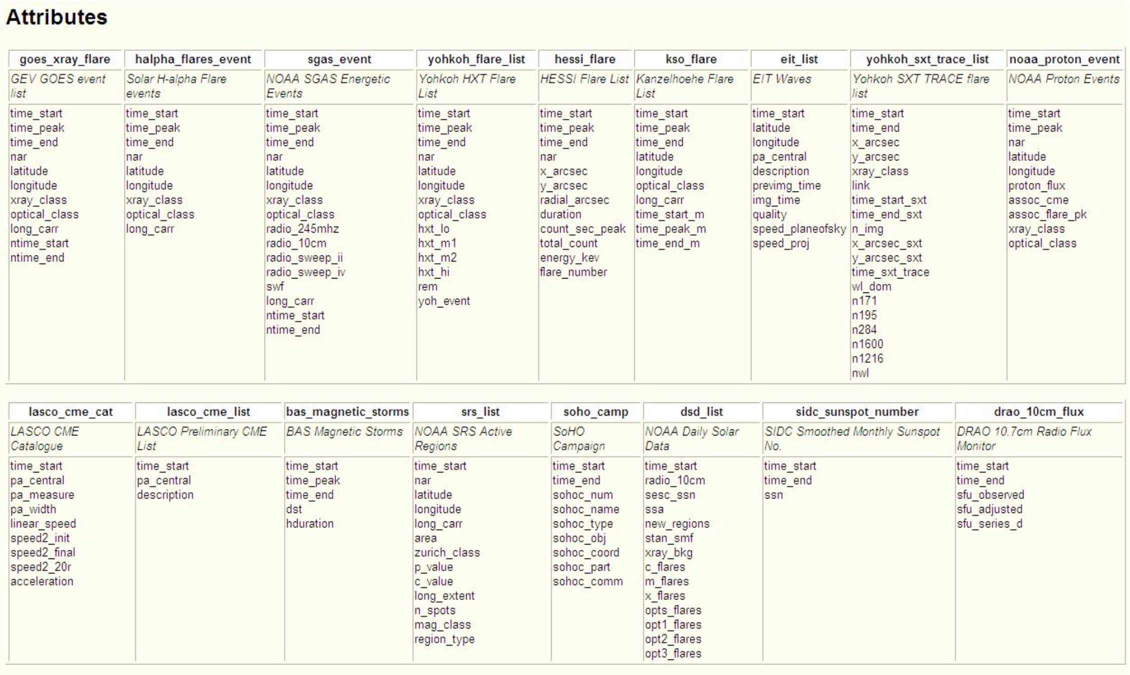
The list of Table names associated with each catalogue and the related URL are reported in Table 2. The attributes relevant to each Table are listed in Table 3.



Tables

Table name	Description	Catalogue URL
goes_xray_flare	GEV GOES event list	ftp://ftp.ngdc.noaa.gov/STP/SOLAR_DATA/SOLAR_FLARES/XRAY_FLARES
halpha_flares_event	Solar H-alpha Flare events	http://www.ngdc.noaa.gov/stp/SOLAR/ftpsolarflares.html#halpha
sgas_event	NOAA SGAS Energetic Events	ftp://solar.sec.noaa.gov/pub/forecasts/SGAS
yohkoh_flare_list	Yohkoh HXT Flare List	http://isass1.solar.isas.jaxa.jp/sxt_co/hxt_flare_list.txt
hessi_flare	HESSI Flare List	http://hesperia.gsfc.nasa.gov/ssw/hessi/dbase
kso_flare	Kanzelhoehe Flare List	http://www.kso.ac.at
eit_list	EIT Waves	http://umbra.nascom.nasa.gov/eit #[Contributed by Barbara Thompson (NASA/GSFC)]
yohkoh_sxt_trace_list	Yohkoh SXT TRACE flare list	http://www.lmsal.com/nitta/sxt_trace_flares/list.html
noaa_proton_event	NOAA Proton Events	http://umbra.nascom.nasa.gov/SEP/seps.html
lasco_cme_cat	LASCO CME Catalogue	http://cdaw.gsfc.nasa.gov/CME_list/UNIVERSAL/text_ver/univ_all.txt
lasco_cme_list	LASCO Preliminary CME List	ftp://lasco6.nascom.nasa.gov/pub/lasco/status
bas_magnetic_storms	BAS Magnetic Storms	http://www.antarctica.ac.uk/SatelliteRisks #[Contributed by Richard Horne (BAS)]
srs_list	NOAA SRS Active Regions	ftp://solar.sec.noaa.gov/pub/forecasts/SRS
soho_camp	SoHO Campaign	http://sohowww.nascom.nasa.gov/data/summary/asplanned/campaign/soho_campaign.dat
dsd_list	NOAA Daily Solar Data	http://www.sec.noaa.gov/ftpdir/indices/old_indices
sidc_sunspot_number	SIDC Smoothed Monthly Sunspot No.	http://sidc.oma.be/DATA/monthssn.dat
drao_10cm_flux	DRAO 10.7cm Radio Flux Monitor	http://www.spaceweather.gc.ca/sx-eng.php

Table 2 List of Table names



Attributes

goes_xray_flare	halpha_flares_event	sgas_event	yohkoh_flare_list	hessi_flare	kso_flare	eit_list	yohkoh_sxt_trace_list	noaa_proton_event
GEV GOES event list	*Solar H-alpha Flare events*	*NOAA SGAS Energetic Events*	*Yohkoh HXT Flare List*	*HESSI Flare List*	*Kanzelhoehe Flare List*	*EIT Waves*	*Yohkoh SXT TRACE flare list*	*NOAA Proton Events*
time_start time_peak time_end nar latitude longitude xray_class optical_class long_carr ntime_start ntime_end	time_start time_peak time_end nar latitude longitude xray_class optical_class long_carr	time_start time_peak time_end nar latitude longitude xray_class optical_class radio_245mhz radio_10cm radio_sweep_ii radio_sweep_iv swf long_carr ntime_start ntime_end	time_start time_peak time_end nar latitude longitude xray_class optical_class hxt_lo hxt_m1 hxt_m2 hxt_hi rem yoh_event	time_start time_peak time_end nar x_arcsec y_arcsec radial_arcsec duration count_sec_peak total_count energy_kev flare_number	time_start time_peak time_end latitude longitude optical_class long_carr time_start_m time_peak_m time_end_m	time_start latitude longitude pa_central description previmg_time img_time quality speed_planeofsky speed_proj	time_start time_end x_arcsec y_arcsec xray_class link time_start_sxt time_end_sxt n_img x_arcsec_sxt y_arcsec_sxt time_sxt_trace wl_dom n171 n195 n284 n1600 n1216 nwl	time_start time_peak nar latitude longitude proton_flux assoc_cme assoc_flare_pk xray_class optical_class

lasco_cme_cat	lasco_cme_list	bas_magnetic_storms	srs_list	soho_camp	dsd_list	sidc_sunspot_number	drao_10cm_flux
LASCO CME Catalogue	*LASCO Preliminary CME List*	*BAS Magnetic Storms*	*NOAA SRS Active Regions*	*SoHO Campaign*	*NOAA Daily Solar Data*	*SIDC Smoothed Monthly Sunspot No.*	*DRAO 10.7cm Radio Flux Monitor*
time_start pa_central pa_measure pa_width linear_speed speed2_init speed2_final speed2_20r acceleration	time_start pa_central description	time_start time_peak time_end dst hduration	time_start nar latitude longitude long_carr area zurich_class p_value c_value long_extent n_spots mag_class region_type	time_start time_end sohoc_num sohoc_name sohoc_type sohoc_obj sohoc_coord sohoc_part sohoc_comm	time_start radio_10cm sesc_ssn ssa new_regions stan_smf xray_bkg c_flares m_flares x_flares opts_flares opt1_flares opt2_flares opt3_flares	time_start time_end ssn	time_start time_end sfu_observed sfu_adjusted sfu_series_d

Table 3 List of attributes

3.1 Catalogue Data and Metadata Tables

The creation of catalogue data and metadata tables is carried out by the **sec_create.sql** procedure as follows:

```
-- EGSO - SEC server
-- # INAF - Trieste Astronomical Observatory
-- creation script
-- by M.Jurcev,A.Santin last rev. 09-May-2005

-- User creation
DROP USER apache;
CREATE USER apache;
DROP USER root;
CREATE USER root CREATEUSER;

-- Creation of catalogue tables
DROP TABLE sgas_event;
CREATE TABLE sgas_event (
      sgs_id                  SERIAL,
      ntime_start       TIMESTAMP,
      time_start        TIMESTAMP,
      time_peak         TIMESTAMP,
      time_end          TIMESTAMP,
      ntime_end         TIMESTAMP,
      nar                     INTEGER,
      latitude          FLOAT,
      longitude         FLOAT,
      long_carr         FLOAT,
      xray_class        VARCHAR(10),
      optical_class     VARCHAR(10),
      radio_245mhz      INTEGER,
      radio_10cm        INTEGER,
      radio_sweep_ii    BOOLEAN,
      radio_sweep_iv    BOOLEAN,
      swf                     VARCHAR(10),
      PRIMARY KEY (sgs_id)
);

DROP TABLE noaa_proton_event;
CREATE TABLE noaa_proton_event (
      npe_id                  SERIAL,
      time_start        TIMESTAMP,
      time_peak         TIMESTAMP,
      nar                     INTEGER,
      latitude          FLOAT,
      longitude         FLOAT,
      proton_flux       FLOAT,
      assoc_cme         VARCHAR(20),
      assoc_flare_pk    TIMESTAMP,
      xray_class        VARCHAR(10),
      optical_class     VARCHAR(10),
      PRIMARY KEY (npe_id)
);

DROP TABLE hessi_flare;
CREATE TABLE hessi_flare (
      hef_id                  SERIAL,
      time_start        TIMESTAMP,
      time_peak         TIMESTAMP,
      time_end          TIMESTAMP,
      nar                     INTEGER,
      x_arcsec          FLOAT,
      y_arcsec          FLOAT,
      radial_arcsec     FLOAT,
      duration          INTEGER,
      count_sec_peak    INTEGER,
      total_count       INTEGER,
      energy_kev        INTEGER,
      flare_number      INTEGER,
      PRIMARY KEY (hef_id)
);

DROP TABLE lasco_cme_list;
CREATE TABLE lasco_cme_list (
      lcl_id            SERIAL,
      time_start  TIMESTAMP,
      pa_central  FLOAT,
      description VARCHAR(512),
      PRIMARY KEY (lcl_id)
);

DROP TABLE lasco_cme_cat;
CREATE TABLE lasco_cme_cat (
      lcc_id                  SERIAL,
      time_start        TIMESTAMP,
      pa_central        FLOAT,
      pa_measure        FLOAT,
      pa_width          FLOAT,
      linear_speed      FLOAT,
      speed2_init       FLOAT,
      speed2_final      FLOAT,
      speed2_20r        FLOAT,
      acceleration      FLOAT,
      PRIMARY KEY (lcc_id)
);

DROP TABLE sidc_sunspot_number;
CREATE TABLE sidc_sunspot_number (
      ssn_id            SERIAL,
      time_start  TIMESTAMP,
      time_end    TIMESTAMP,
      ssn               FLOAT,
      PRIMARY KEY (ssn_id)
);

DROP TABLE drao_10cm_flux;
CREATE TABLE drao_10cm_flux (
      sfm_id                  SERIAL,
      time_start        TIMESTAMP,
      time_end          TIMESTAMP,
      sfu_observed      FLOAT,
      sfu_adjusted      FLOAT,
      sfu_series_d      FLOAT,
      PRIMARY KEY (sfm_id)
);

DROP TABLE dsd_list;
CREATE TABLE dsd_list (
      dsd_id            SERIAL,
      time_start        TIMESTAMP,
      radio_10cm        INTEGER,
      sesc_ssn          INTEGER,
      ss_area                 INTEGER,
      new_regions       INTEGER,
      stan_smf          INTEGER,
      xray_bkg          VARCHAR(10),
      c_flares          INTEGER,
      m_flares          INTEGER,
      x_flares          INTEGER,
      opts_flares       INTEGER,
      opt1_flares       INTEGER,
      opt2_flares       INTEGER,
      opt3_flares       INTEGER,
      PRIMARY KEY (dsd_id)
);

DROP TABLE yohkoh_flare_list;
CREATE TABLE yohkoh_flare_list (
      yfc_id                  SERIAL,
      time_start        TIMESTAMP,
      time_peak         TIMESTAMP,
      time_end          TIMESTAMP,
      nar                     INTEGER,
      latitude          FLOAT,
      longitude         FLOAT,
      xray_class        VARCHAR(10),
      optical_class     VARCHAR(10),
      hxt_lo                  INTEGER,
      hxt_m1                  INTEGER,
      hxt_m2                  INTEGER,
      hxt_hi                  INTEGER,
      rem                     VARCHAR(3),
      yoh_event         INTEGER,
      PRIMARY KEY (yfc_id)
);

DROP TABLE srs_list;
CREATE TABLE srs_list (
      srs_id                  SERIAL,
      time_start        TIMESTAMP,
      nar                     INTEGER,
      latitude          FLOAT,
      longitude         FLOAT,
      long_carr         FLOAT,
      area              FLOAT,
      zurich_class      VARCHAR(2),
      p_value                 VARCHAR(2),
      c_value                 VARCHAR(2),
      long_extent       FLOAT,
      n_spots                 INTEGER,
      mag_class         VARCHAR(20),
      region_type       VARCHAR(30),
      PRIMARY KEY (srs_id)
);

DROP TABLE bas_magnetic_storms;
CREATE TABLE bas_magnetic_storms (
      bms_id                  SERIAL,
      time_start        TIMESTAMP,
      time_peak         TIMESTAMP,
      time_end          TIMESTAMP,
      dst                     INTEGER,
      hduration         INTEGER,
      PRIMARY KEY (bms_id)
);

DROP TABLE goes_xray_flare;
CREATE TABLE goes_xray_flare (
      goes_id                 SERIAL,
      ntime_start       TIMESTAMP,
      time_start        TIMESTAMP,
      time_peak         TIMESTAMP,
      time_end          TIMESTAMP,
      ntime_end         TIMESTAMP,
    nar                       INTEGER,
      latitude          FLOAT,
    longitude           FLOAT,
      long_carr         FLOAT,
      xray_class        VARCHAR(10),
    optical_class VARCHAR(10),
    PRIMARY KEY (goes_id));

DROP TABLE soho_camp;
CREATE TABLE soho_camp (
      soho_id                 SERIAL,
      time_start        TIMESTAMP,
      time_end          TIMESTAMP,
    sohoc_num           INTEGER,
    sohoc_name    VARCHAR(128),
    sohoc_type    VARCHAR(384),
    sohoc_obj     VARCHAR(2048),
    sohoc_coord   VARCHAR(128),
    sohoc_part    VARCHAR(3328),
    sohoc_comm    VARCHAR(384),
    PRIMARY KEY (soho_id)
);

DROP TABLE kso_flare;
CREATE TABLE kso_flare (
      kso_id                  SERIAL,
      time_start        TIMESTAMP,
      time_start_m      VARCHAR(2),
      time_peak         TIMESTAMP,
      time_peak_m       VARCHAR(2),
      time_end          TIMESTAMP,
      time_end_m      VARCHAR(2),
      latitude          FLOAT,
    longitude           FLOAT,
      long_carr         FLOAT,
    optical_class VARCHAR(10),
    PRIMARY KEY (kso_id)
);

DROP TABLE eit_list;
CREATE TABLE eit_list (
      eit_id                  SERIAL,
      time_start        TIMESTAMP,
      previmg_time            TIMESTAMP,
      img_time          TIMESTAMP,
      quality                 VARCHAR(4),
      latitude          FLOAT,
      longitude         FLOAT,
      speed_planeofsky        FLOAT,
    speed_proj        FLOAT,
    pa_central        FLOAT,
      description       VARCHAR(128),
      PRIMARY KEY (eit_id)
);

DROP TABLE yohkoh_sxt_trace_list;
CREATE TABLE yohkoh_sxt_trace_list (
      yst_id                  SERIAL,
    link    VARCHAR(128),
      time_start_sxt          TIMESTAMP,
      time_end_sxt            TIMESTAMP,
      xray_class        VARCHAR(10),
      n_img       INTEGER,
      x_arcsec_sxt            FLOAT,
      y_arcsec_sxt            FLOAT,
      time_start        TIMESTAMP,
      time_end          TIMESTAMP,
      time_sxt_trace    TIMESTAMP,
    wl_dom  INTEGER,
    x_arcsec            FLOAT,
      y_arcsec          FLOAT,
      n171        INTEGER,
      n195        INTEGER,
      n284        INTEGER,
      n1600       INTEGER,
      n1216       INTEGER,
      nwl         INTEGER,
      PRIMARY KEY (yst_id)
);

DROP TABLE halpha_flares_event;
CREATE TABLE halpha_flares_event (
      ha_id             SERIAL,
      time_start        TIMESTAMP,
      time_peak         TIMESTAMP,
      time_end          TIMESTAMP,
    nar                       INTEGER,
      latitude          FLOAT,
    longitude           FLOAT,
      long_carr         FLOAT,
      xray_class        VARCHAR(10),
    optical_class VARCHAR(10),
    PRIMARY KEY (ha_id));


-- Creation of metadata tables
DROP TABLE sec_catalogue;
CREATE TABLE sec_catalogue (
      cat_id                  INTEGER,
      name              VARCHAR(30),
      description       VARCHAR(80),
      type              VARCHAR(20),
      status                  VARCHAR(20),
      url                     VARCHAR(80),
      PRIMARY KEY (cat_id)
);
```

```
DROP TABLE sec_attribute;
CREATE TABLE sec_attribute (
      attr_id                 INTEGER,
      name              VARCHAR(30),
      description       VARCHAR(80),
      type              VARCHAR(20),
      PRIMARY KEY (attr_id)
);

DROP TABLE sec_cat_attr;
CREATE TABLE sec_cat_attr (
      cat_id                  INTEGER,
      attr_id                 INTEGER
);


-- Permission settings
REVOKE ALL ON TABLE
      sec_catalogue,
      sec_attribute,
      sec_cat_attr,
      noaa_proton_event,
      sgas_event,
      hessi_flare,
      lasco_cme_list,
      lasco_cme_cat,
      sidc_sunspot_number,
      drao_10cm_flux,
      dsd_list,
      yohkoh_flare_list,
      srs_list,
      bas_magnetic_storms,
      goes_xray_flare,
      soho_camp,
    kso_flare,
    eit_list,
    yohkoh_sxt_trace_list,
    halpha_flares_event
FROM apache;

GRANT SELECT ON TABLE
      sec_catalogue,
      sec_attribute,
      sec_cat_attr,
      noaa_proton_event,
      sgas_event,
      hessi_flare,
      lasco_cme_list,
      lasco_cme_cat,
      sidc_sunspot_number,
      drao_10cm_flux,
      dsd_list,
      yohkoh_flare_list,
      srs_list,
      bas_magnetic_storms,
      goes_xray_flare,
      soho_camp,
    kso_flare,
    eit_list,
    yohkoh_sxt_trace_list,
    halpha_flares_event
```

```
TO apache;

REVOKE ALL ON TABLE
      sec_catalogue,
      sec_attribute,
      sec_cat_attr,
      noaa_proton_event,
      sgas_event,
      hessi_flare,
      lasco_cme_list,
      lasco_cme_cat,
      sidc_sunspot_number,
      drao_10cm_flux,
      dsd_list,
      yohkoh_flare_list,
      srs_list,
      bas_magnetic_storms,
      goes_xray_flare,
      soho_camp,
    kso_flare,
    eit_list,
    yohkoh_sxt_trace_list,
    halpha_flares_event
FROM root;

GRANT ALL ON TABLE
      sec_catalogue,
      sec_attribute,
      sec_cat_attr,
      noaa_proton_event,
      sgas_event,
      hessi_flare,
      lasco_cme_list,
      lasco_cme_cat,
      sidc_sunspot_number,
      drao_10cm_flux,
      dsd_list,
      yohkoh_flare_list,
      srs_list,
      bas_magnetic_storms,
      goes_xray_flare,
      soho_camp,
    kso_flare,
    eit_list,
    yohkoh_sxt_trace_list,
    halpha_flares_event
TO root;
```

3.2 Data Attributes

The data attributes are detailed by the **sec_fillmetadata.sql** procedure, which lists name, meaning and type of all the unique table fields as follows:

```
-- attribute description
-- types:
--          t = datetime
--          i = integer number
--          f = float number
--          b = boolean
--          s = string
DELETE FROM sec_attribute;
1,'time_start',         'event start time',                 't');
2,'time_peak',          'event peak time',                  't');
3,'time_end',           'event end time',                   't');
4,'nar',                'active region number',             'i');
5,'latitude',           'heliographic latitude',            'f');
6,'longitude',          'heliographic longitude',           'f');
7,'x_arcsec',           'x position [arcsec]'   ,           'f');
8,'y_arcsec',           'y position [arcsec]',              'f');
9,'radial_arcsec',      'sun radius [arcsec]',              'f');
10,'pa_central',        'polar angle [deg]',                'f');
11,'pa_measure',        'polar angle measure [deg]',        'f');
12,'proton_flux',       'proton flux units',                'f');
13,'assoc_cme',         'associated CME',                   's');
14,'assoc_flare_pk',    'peak time for associated flare',   't');
15,'xray_class',        'x-ray importance class',           's');
16,'optical_class',     'optical importance class',         's');
17,'radio_245mhz',      'radio emission at 245 MHz',        'i');
18,'radio_10cm',        'radio emission at 10 cm',          'i');
19,'radio_sweep_ii',    'radio sweep type II',              'b');
20,'radio_sweep_iv',    'radio sweep type IV',              'b');
21,'swf',               'short wave fade',                  's');
22,'duration',          'duration [s]',                     'i');
23,'count_sec_peak',    'peak of counts per sec [s-1]',     'i');
24,'total_count',       'total counts',                     'i');
25,'energy_kev',        'energy [keV]',                     'f');
26,'flare_number',      'flare id number',                  'i');
27,'description',       'event description',                's');
28,'pa_width',          'polar angle width [deg]',          'f');
29,'linear_speed',      'linear speed [km/sec]',            'f');
30,'speed2_init',       '2nd order initial speed [km/sec]', 'f');
31,'speed2_final',      '2nd order final speed [km/sec]',   'f');
32,'speed2_20r',        '2nd order speed at 20 R [km/sec]', 'f');
33,'acceleration',      'acceleration [km/sec^2]',          'f');
34,'ssn',               'sunspot number',                   'f');
35,'sfu_observed',      'observed solar flux units',        'f');
36,'sfu_adjusted',      'adjusted solar flux units',        'f');
37,'sfu_series_d',      'series d solar flux units',        'f');
38,'sesc_ssn',          'sesc sunspot number',              'i');
39,'ssa',               'sunspot area',                     'i');
40,'new_regions',       'new regions',                      'i');
41,'stan_smf',          'stan smf',                         'i');
42,'xray_bkg',          'x-ray bkg',                        's');
43,'c_flares',          'class C flares',                   'i');
44,'m_flares',          'class M flares',                   'i');
45,'x_flares',          'class X flares',                   'i');
46,'opts_flares',       'opts flares',                      'i');
47,'opt1_flares',       'opt1 flares',                      'i');
48,'opt2_flares',       'opt2 flares',                      'i');
49,'opt3_flares',       'opt3 flares',                      'i');
50,'hxt_lo',            'HXT lo',                           'i');
51,'hxt_m1',            'HXT m1',                           'i');
52,'hxt_m2',            'HXT m2',                           'i');
53,'hxt_hi',            'HXT hi',                           'i');
54,'rem',               'Remarks',                          's');
55,'yoh_event',         'Yohkoh event',                     'i');
56,'long_carr',         'Carrington longitude',             'f');
57,'area',              'area',                             'f');
58,'zurich_class',      'Zurich classification',            's');
59,'p_value',           'p value',                          's');
```

```
60,'c_value',           'c value',                          's');
61,'long_extent',       'longitudinal extension',           'f');
62,'n_spots',           'number of sunspots',               'i');
63,'mag_class',         'mag class',                        's');
64,'region_type',       'active region type',               's');
65,'dst',               'magnetic field [nT]',              'i');
66,'hduration',         'duration [h]',                     'i');
67,'sohoc_num',         'SoHO Campaign number',             'i');
68,'sohoc_name',        'SoHO Campaign name',               's');
69,'sohoc_type',        'SoHO Campaign type',               's');
70,'sohoc_obj',         'SoHO Campaign objective',          's');
71,'sohoc_coord',       'SoHO Campaign coordinators',       's');
72,'sohoc_part',        'SoHO Campaign participants',       's');
73,'sohoc_comm',        'SoHO Campaign comments',           's');
74,'ntime_start',       'selection start time',             't');
75,'ntime_end',         'selection end time',               't');
76,'time_start_m',      'event start time modifier',        's');
77,'time_peak_m',       'event peak modifier',              's');
78,'time_end_m',        'event end time modifier',          's');
79,'previmg_time',      'Previous Image Time',              't');
80,'img_time',          'Image Time',                       't');
81,'quality',           'Quality Rating',                   's');
82,'speed_planeofsky',  'Plane-of-Sky Speed',               'f');
83,'speed_proj',        'Projected Speed',                  'f');
84,'link',              'The location of the SXT PFI file at ISAS','s');
85,'time_start_sxt',    'Times of the first flare mode SXT images','t');
86,'time_end_sxt',      'Times of the last flare mode SXT images', 't');
87,'n_img',             'Number of flare mode SXT images',  'i');
88,'x_arcsec_sxt',      'Coord. X (arcsec from disk center) of the
                        center of the 6th SXT flare mode image',   'f');
89,'y_arcsec_sxt',      'Coord. Y (arcsec from disk center) of the center
                        of the 6th SXT flare mode image',   'f');
90,'time_sxt_trace',    'Time of the TRACE image closest to
                        time_start_sxt',                           't');
91,'wl_dom',            'Wavelength at which the greatest number of
                  images is taken',      'i');
92,'n171',        'Number of TRACE images at 171 A',        'i');
93,'n195',        'Number of TRACE images at 195 A',        'i');
94,'n284',        'Number of TRACE images at 284 A',        'i');
95,'n1600',       'Number of TRACE images at 1600 A',       'i');
96,'n1216',       'Number of TRACE images at 1216 A',       'i');
97,'nwl',   'Number of TRACE images at WL',      'i');
```

3.3 Catalogue-Data Attribute Cross-Matrix

The attribute-catalogue relationship is schematized in Figure 3.

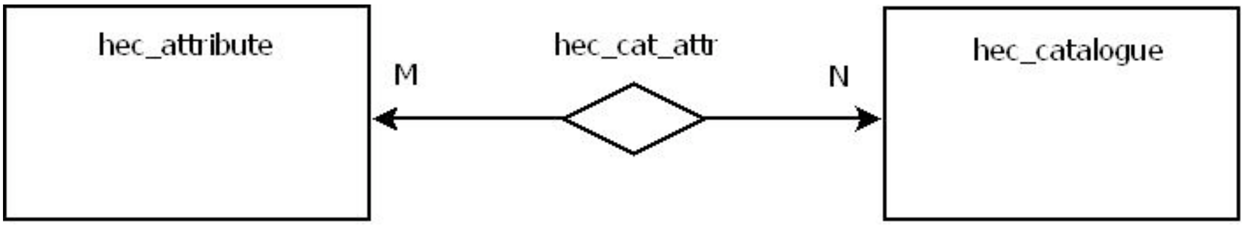
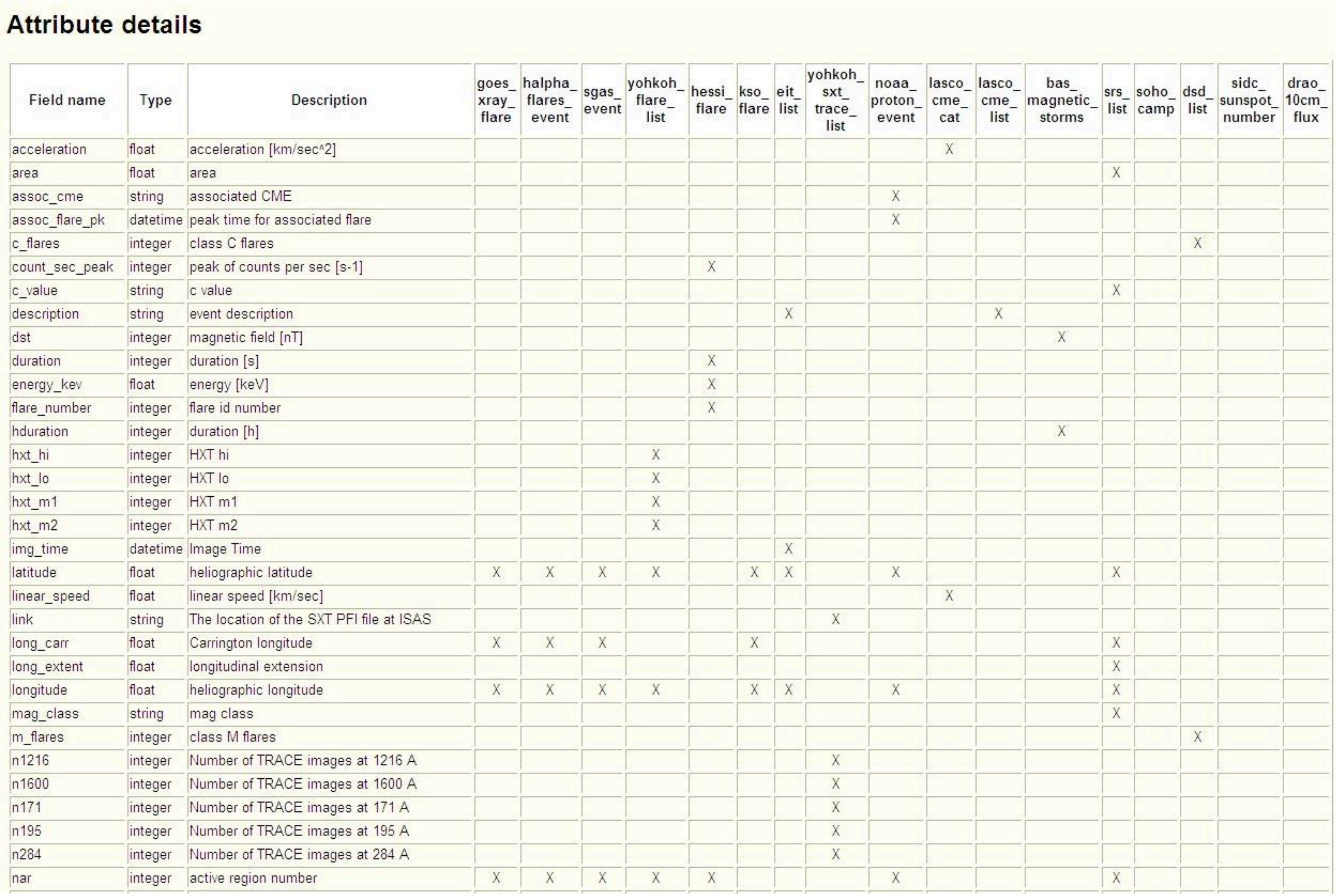


Figure 3 Attribute-catalogue relationship schema

Table 4 excerpted illustrates the correspondence (cross-matrix) between all the catalogue providers available data (columns) and all the above attributes (rows).

Each row of the cross-matrix shows all the available catalogues offering a certain attribute, while every column gives all the available data fields of a single provider.



Attribute details

Field name	Type	Description	goes_xray_flare	halpha_flares_event	sgas_event	yohkoh_flare_list	hessi_flare	kso_flare	eit_list	yohkoh_sxt_trace_list	noaa_proton_event	lasco_cme_cat	lasco_cme_list	bas_magnetic_storms	srs_list	soho_camp	dsd_list	sidc_sunspot_number	drao_10cm_flux
acceleration	float	acceleration [km/sec^2]										X							
area	float	area													X				
assoc_cme	string	associated CME									X								
assoc_flare_pk	datetime	peak time for associated flare									X								
c_flares	integer	class C flares															X		
count_sec_peak	integer	peak of counts per sec [s-1]					X												
c_value	string	c value													X				
description	string	event description							X				X						
dst	integer	magnetic field [nT]												X					
duration	integer	duration [s]					X												
energy_kev	float	energy [keV]					X												
flare_number	integer	flare id number					X												
hduration	integer	duration [h]												X					
hxt_hi	integer	HXT hi				X													
hxt_lo	integer	HXT lo				X													
hxt_m1	integer	HXT m1				X													
hxt_m2	integer	HXT m2				X													
img_time	datetime	Image Time							X										
latitude	float	heliographic latitude	X	X	X	X		X	X		X				X				
linear_speed	float	linear speed [km/sec]										X							
link	string	The location of the SXT PFI file at ISAS								X									
long_carr	float	Carrington longitude	X	X	X			X							X				
long_extent	float	longitudinal extension													X				
longitude	float	heliographic longitude	X	X	X	X		X	X		X				X				
mag_class	string	mag class													X				
m_flares	integer	class M flares															X		
n1216	integer	Number of TRACE images at 1216 A								X									
n1600	integer	Number of TRACE images at 1600 A								X									
n171	integer	Number of TRACE images at 171 A								X									
n195	integer	Number of TRACE images at 195 A								X									
n284	integer	Number of TRACE images at 284 A								X									
nar	integer	active region number	X	X	X	X	X				X				X				

Table 4 Catalogue-data attribute cross-matrix

3.4 Function Details

The system cron service activates the download from catalogue providers every day at 04.00

UTC.

Function calls tree

cron -> sec_loader.sh

sec_loader.sh

	catalogue load and parse functions
sec_load_hef.php	HESSI Flare List
sec_load_npe.php	NOAA Proton Events
sec_load_lcl.php	LASCO Preliminary CME List
sec_load_lcc.php	LASCO CME Catalogue
sec_load_ssn.php	SIDC Smoothed Monthly Sunspot No.
sec_load_sfm.php	DRAO 10.7cm Radio Flux Monitor
sec_load_dsd.php	NOAA Daily Solar Data
sec_load_sgs.php	Solar H-alpha Flare events
sec_load_yfc.php	Yohkoh HXT Flare List
sec_load_srs.php	NOAA SRS Active Regions
sec_load_bms.php	BAS Magnetic Storms
sec_load_goes.php	GEV GOES event list
sec_load_soho.php	SoHO Campaign
sec_load_kso.php	Kanzelhoehe Flare List
sec_load_eit.php	EIT Waves
sec_load_yst.php	Yohkoh SXT TRACE Flare List
sec_load_ha.php	Solar H-alpha Flare events
sec_create.sql	database structure creation
sec_fillmetadata.sql	metadata tables popolation
sec_doc_gen.php	HTML documentation from metadata generation
sec_insert.sql	catalogues tables population
sec_range.php	HTML catalogue description (date range, # records)
sec_graph.php	PNG graph list entry population

- **sec_loader.sh** calls all the php functions which download and parse downloaded data from all the catalogues.

Such php functions write temporary converted data files in the temp/ directory and these files are used by the procedure **sec_insert.sql** to populate/update the PostgreSQL database tables (as detailed in **sec_create.sql** procedure).

The general scheme of a loader procedure is given in the following:

```
<?php
      # ============================================
      # EGSO 2003 SEC server - by Max Jurcev, Andrej Santin
      # INAF - Trieste Astronomical Observatory
      # --------------------------------------------
      # sec_load_template.php
      # 1st 26-nov-03, last 26-nov-03
      # =============================================
      require ("sec_global.php");

      // get files from HTTP or FTP
      // copy files to temporary directory
      // delete local file if needed

      // parse files and create postgres-ready file   (\t   tab   separated
values)
      $f1 = fopen("$tempdir/XXX.postgres.converted",'w');
      $f2 = fopen("$tempdir/data.dat",'r');
      while (!feof ($f2)) {
            //get one row
    $buffer = fgets($f2);

    //extract needed values
    $year = substr($buffer,0,4);
            $month = substr($buffer,4,2);
            $day = substr($buffer,6,2);
            $x = substr($buffer,9,2);
            if ($x == "") $x = "\N";//empty values MUST be filled with a \N
null

    //write to postgres-ready file
            $out    =    sprintf("%04d-%02d-01    00:00:00\t%04d-%02d-%02d
23:59:59\t%f\n",$year,$month,$year,$month,$day,$x);
            fwrite($f1,$out);
      }//while

      fclose($f2);
      fclose($f1);

?>
```

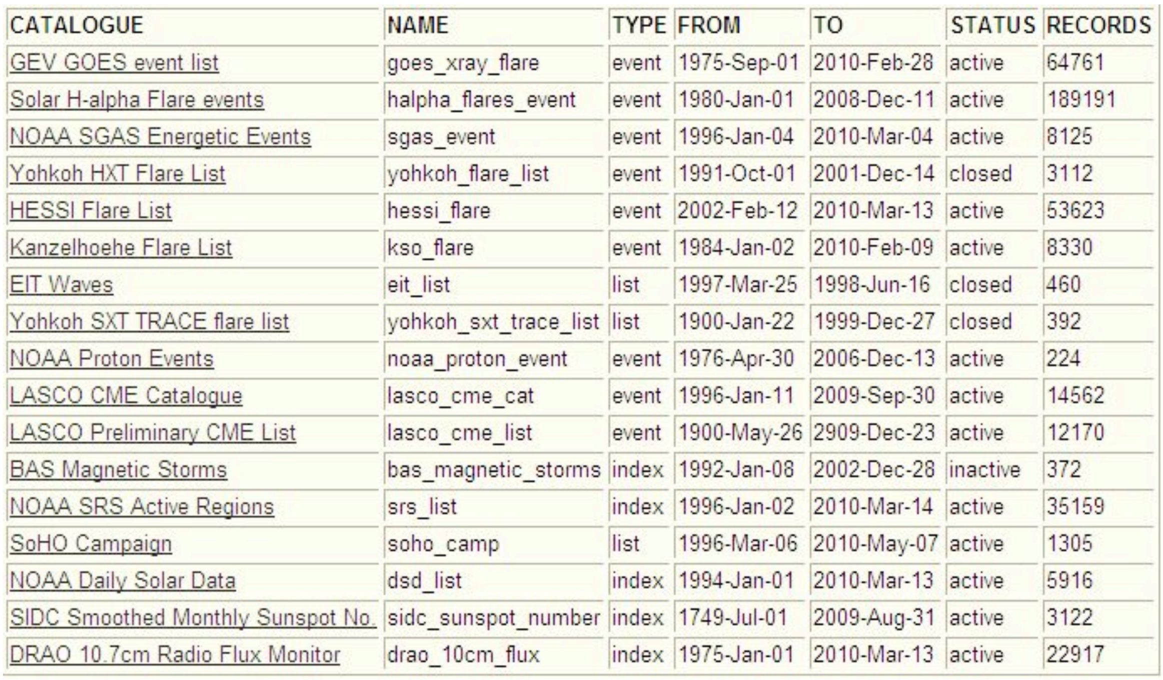
As an example the listing of the **sec_load_goes.php** procedure that loads and parses the GEV GOES event list is reported in the following:

```
<?php
      # =============================================
      # EGSO 2003 SEC server - by Andrej Santin
      # INAF - Trieste Astronomical Observatory
      # sec_load_goes.php
      # 06-jul-2004, last 12-oct-2004
      # ---------------------------------------------
      # read the GOES xray flare data
      #
ftp://anonymous@ftp.ngdc.noaa.gov/STP/SOLAR_DATA/SOLAR_FLARES/XRAY_FLARES
      # =============================================
      define("DEBUG","0");
      set_time_limit(60);//may take a long time...

      require ("sec_global.php");


      // get files from HTTP
      if (DEBUG==0) {
            exec ("/bin/rm temp/goes/xray*");
            exec ("/bin/rm temp/goes/XRAY*");
            exec ("wget -P temp/goes -N -A 'XRAY[1-9]*','xray[1-9]*'
ftp://anonymous@ftp.ngdc.noaa.gov/STP/SOLAR_DATA/SOLAR_FLARES/XRAY_FLARES/\
*");
            exec ("cat temp/goes/xray* temp/goes/XRAY* > all_xray.txt");
            copy ("all_xray.txt",$tempdir."/all_xray.txt");
            unlink ("all_xray.txt");
      }

      // parse files and create postgres-ready file
      $f1 = fopen("$tempdir/GOES.postgres.converted",'w');
      $f2 = fopen("$tempdir/all_xray.txt",'r');
      $ntime_err=0;

      while (!feof ($f2)) {
            $i=1; //buf array index
            $buffer = fgets($f2);
            $buffer=substr($buffer,1,255);//echo "?????????\n";
            // get date
            $y = substr($buffer,5,2);
            if ($y<70) $y=$y+2000; else $y=$y+1900;
            $mo = substr($buffer,7,2);
            $d = substr($buffer,9,2);
            $jd = GregorianToJD ($mo,$d,$y);
            //next day
            $gregorian = JDToGregorian ($jd+1);// mm/dd/yyyy
            $dd=split("/",$gregorian);
            $dateinc=sprintf("%04d/%02d/%02d",$dd[2],$dd[0],$dd[1]);// next
day date
            //day before
            $gregorian = JDToGregorian ($jd-1);// mm/dd/yyyy
            $dd=split("/",$gregorian);
            $datedec=sprintf("%04d/%02d/%02d",$dd[2],$dd[0],$dd[1]);//
previous day date
            //next two day
            $gregorian = JDToGregorian ($jd+2);// mm/dd/yyyy
            $dd=split("/",$gregorian);
            $dateinc2=sprintf("%04d/%02d/%02d",$dd[2],$dd[0],$dd[1]);//
next day date
            //next three day
            $gregorian = JDToGregorian ($jd+2);// mm/dd/yyyy
            $dd=split("/",$gregorian);
            $dateinc3=sprintf("%04d/%02d/%02d",$dd[2],$dd[0],$dd[1]);//
next day date
            if (checkdate($mo,$d,$y)) {
            $date = sprintf("%04d/%02d/%02d",$y,$mo,$d);
        } else {
            $date = "";
        }

        //start time
        if (strlen(trim(substr($buffer,13,4)))==4) {

$m_start=substr($buffer,13,2)*60+substr($buffer,15,2);//start time in
minutes
            if (substr($buffer,13,2)<24) {
                $buf[$i++]=$date."
".substr($buffer,13,2).":".substr($buffer,15,2).":00";
                        $et=substr($buffer,13,2)+substr($buffer,15,2)/60;
                  } else {
                $buf[$i++]=$dateinc." ".(substr($buffer,13,2)-
24).":".substr($buffer,15,2).":00";
```

```
                        $et=(substr($buffer,13,2)-
24)+substr($buffer,15,2)/60;
                  }
        } else {
            $buf[$i++]="";
                  $et=-1;
        }
        //end time
        if (strlen(trim(substr($buffer,18,4)))==4) {
                  $m_end=substr($buffer,18,2)*60+substr($buffer,20,2);//end
time in minutes
                  if (substr($buffer,18,2)>=24) // dateinc,h=h-24
              $buf[$i++]=$dateinc." ".(substr($buffer,18,2)-
24).":".substr($buffer,20,2).":00";
                  else {
                if ($m_start>$m_end) // dateinc,h=h
                    $buf[$i++]=$dateinc."
".(substr($buffer,18,2)).":".substr($buffer,20,2).":00";
                      else {
                    $buf[$i++]=$date."
".substr($buffer,18,2).":".substr($buffer,20,2).":00";
                      }
            }
        } else {
            $buf[$i++]="";
        }
        //peak time
        if (strlen(trim(substr($buffer,23,4)))==4) {

      $m_peak=substr($buffer,23,2)*60+substr($buffer,25,2);//peak time in
minutes
                  if (substr($buffer,23,2)>=24) // dateinc,h=h-24
              $buf[$i++]=$dateinc." ".(substr($buffer,23,2)-
24).":".substr($buffer,25,2).":00";
                  else {
                if ($m_start>$m_peak) // dateinc,h=h
                    $buf[$i++]=$dateinc."
".(substr($buffer,23,2)).":".substr($buffer,25,2).":00";
                else {
                    $buf[$i++]=$date."
".substr($buffer,23,2).":".substr($buffer,25,2).":00";
                }
                  }
        } else {
            $buf[$i++]="";
        }

        if ($m_start>$m_peak) { $m_peak+=1440; echo "peakinc:$buffer\n";
$f=1;} else $f=0;
        if ($m_start>$m_end) $m_end+=1440;
            if (($buf[1]!="") and ($buf[2]!="") and ($buf[3]!="")
             and ($m_start<=$m_peak) and ($m_peak<=$m_end)) {
                  //ntime_start *  selection start time (t0) as a full
flare rise time before the declared start time
                  $t0=$m_peak-2*($m_peak-$m_start);
                  $t0_date=$date;
                  if ($t0<0) {
                        $t0+=1440;
                        $t0_date=$datedec;
                  }
                  $t0h=sprintf("%02d",floor($t0/60));
                  $t0m=sprintf("%02d",$t0 % 60);
                  $ntime_start=$t0_date." ".$t0h.":".$t0m.":00";
                  //ntime_end *  selection end time (t1) as 8 times the
time to half intensity after the flare maximum (=> ~1/50th intensity)
                  $t1=$m_peak+8*($m_end-$m_peak);
                  $t1_date=$date;
                  if ($t1>1439) {
                        $t1=$t1-1440;
                        $t1_date=$dateinc;
                  }
                  if ($t1>1439) {
                        $t1=$t1-1440;
                        $t1_date=$dateinc2;
                  }
                  if ($t1>1439) {
                        $t1=$t1-1440;
                        $t1_date=$dateinc3;
                  }
                  $t1h=sprintf("%02d",floor($t1/60));
                  $t1m=sprintf("%02d",$t1 % 60);
                  $ntime_end=$t1_date." ".$t1h.":".$t1m.":00";
            } else {
                  echo ("err:$buffer\n");
            $ntime_err++;
                  $ntime_start="";
                  $ntime_end="";
            }

        //swap end and peak times
        $tmp=$buf[2];
        $buf[2]=$buf[3];
        $buf[3]=$tmp;

        // convert lat
        $st=trim(substr($buffer,28,3));
        if (substr($st,0,1)=="S")
            $buf[$i++]="-".substr($st,1,2);
        else
            $buf[$i++]=substr($st,1,2);
        // convert long
        $st=trim(substr($buffer,31,3));
        if (substr($st,0,1)=="E")
            $buf[$i++]="-".substr($st,1,2);
        else
            $buf[$i++]=substr($st,1,2);
            if (($buf[$i-1]!="") and ($et>-1))
                  $longcarr=long_carr($y,$mo,$d,$et,$buf[$i-1]);
            else
                  $longcarr="";


        // optical class
        $buf[$i++] = strtolower(trim(substr($buffer,34,2)));
        // xray class
            $st=sprintf("%.1f",trim(substr($buffer,60,3))/10);
            $buf[$i++] = substr($buffer,59,1).$st;
        // nar
            $nar = trim(substr($buffer,80,5));// % 10000;
        if (is_numeric($nar)) $buf[$i++] = $nar; else $buf[$i++] =
"";//38684
```

```
              $buf[$i++]=$ntime_start;
              $buf[$i++]=$ntime_end;
              $buf[$i++]=$longcarr;

          if ($buf[1]=="") $out="\N"; else $out=$buf[1];
          for ($k=2;$k<=11;$k++) {
              if ($buf[$k]=="") $buf[$k]="\N";
              $out.="\t".$buf[$k];
          }
          $out.="\n";
          if ($buf[1]!="") fwrite($f1,$out);
        }
        fclose($f2);
        fclose($f1);
    echo "ntime_err=$ntime_err\n";
?>
```

- **sec_insert.sql** populates catalogue tables;
- **sec_fillmetadata.sql** populates metadata tables.
- **sec_doc_gen.php** generates HTML documentation page from metadata tables.
- **sec_range.php** generates the following HTML table (Table 5):



CATALOGUE	NAME	TYPE	FROM	TO	STATUS	RECORDS
GEV GOES event list	goes_xray_flare	event	1975-Sep-01	2010-Feb-28	active	64761
Solar H-alpha Flare events	halpha_flares_event	event	1980-Jan-01	2008-Dec-11	active	189191
NOAA SGAS Energetic Events	sgas_event	event	1996-Jan-04	2010-Mar-04	active	8125
Yohkoh HXT Flare List	yohkoh_flare_list	event	1991-Oct-01	2001-Dec-14	closed	3112
HESSI Flare List	hessi_flare	event	2002-Feb-12	2010-Mar-13	active	53623
Kanzelhoehe Flare List	kso_flare	event	1984-Jan-02	2010-Feb-09	active	8330
EIT Waves	eit_list	list	1997-Mar-25	1998-Jun-16	closed	460
Yohkoh SXT TRACE flare list	yohkoh_sxt_trace_list	list	1900-Jan-22	1999-Dec-27	closed	392
NOAA Proton Events	noaa_proton_event	event	1976-Apr-30	2006-Dec-13	active	224
LASCO CME Catalogue	lasco_cme_cat	event	1996-Jan-11	2009-Sep-30	active	14562
LASCO Preliminary CME List	lasco_cme_list	event	1900-May-26	2909-Dec-23	active	12170
BAS Magnetic Storms	bas_magnetic_storms	index	1992-Jan-08	2002-Dec-28	inactive	372
NOAA SRS Active Regions	srs_list	index	1996-Jan-02	2010-Mar-14	active	35159
SoHO Campaign	soho_camp	list	1996-Mar-06	2010-May-07	active	1305
NOAA Daily Solar Data	dsd_list	index	1994-Jan-01	2010-Mar-13	active	5916
SIDC Smoothed Monthly Sunspot No.	sidc_sunspot_number	index	1749-Jul-01	2009-Aug-31	active	3122
DRAO 10.7cm Radio Flux Monitor	drao_10cm_flux	index	1975-Jan-01	2010-Mar-13	active	22917

Table 5 List of catalogue time spans, activity status and data volume

- **sec_graph.php** generates the following PNG image:

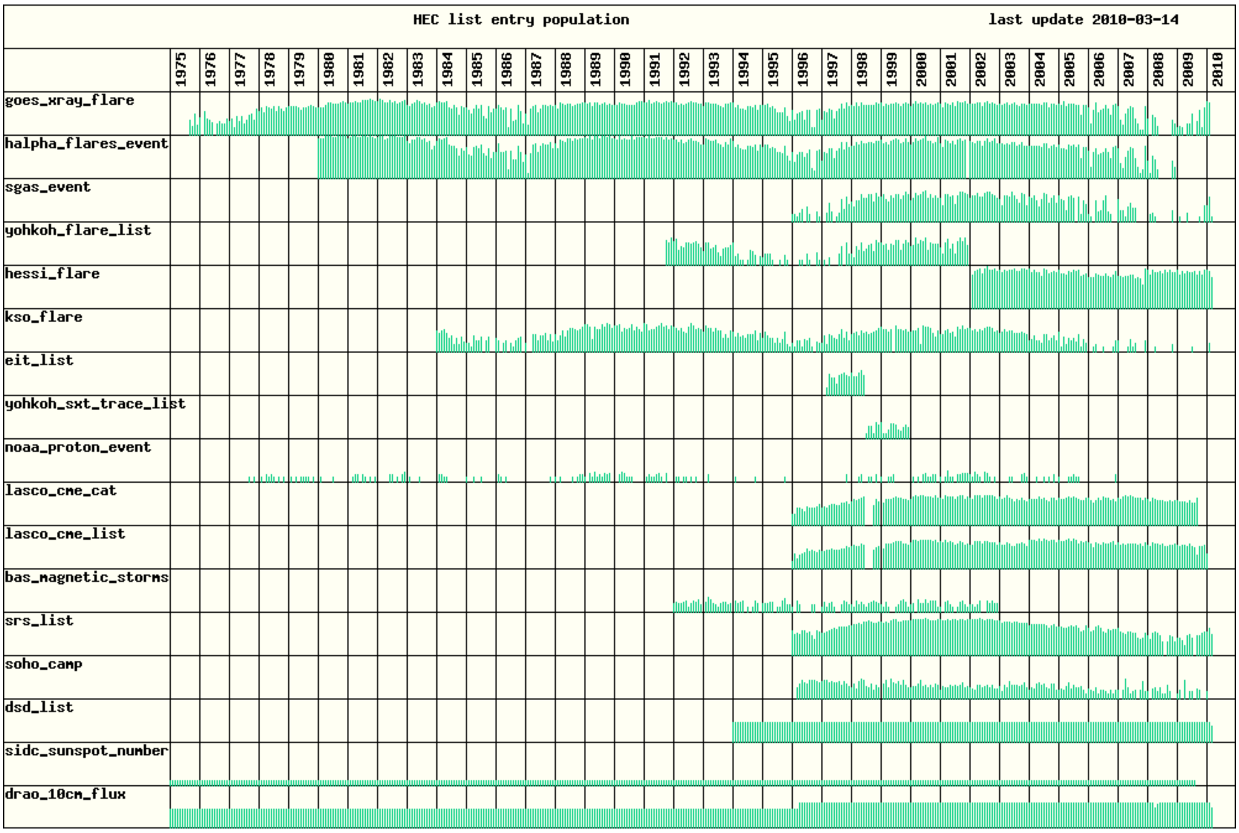


Figure 4 Graph reporting the list entry population

4 Web Service

The HEC Web Service **wsSEC** (see **Figure 5**) publishes one web method named “sql”.

The “sql” method accepts as input a string, formatted as any SQL (SELECT) statement on any table/view defined inside the DB and return a string formatted as a VOTable.

To consume an axis Web Service one needs to generate stubs using Axis WSDL-to-Java tool in "org.apache.axis.wsdl.WSDL2Java".

The basic invocation form looks like this:

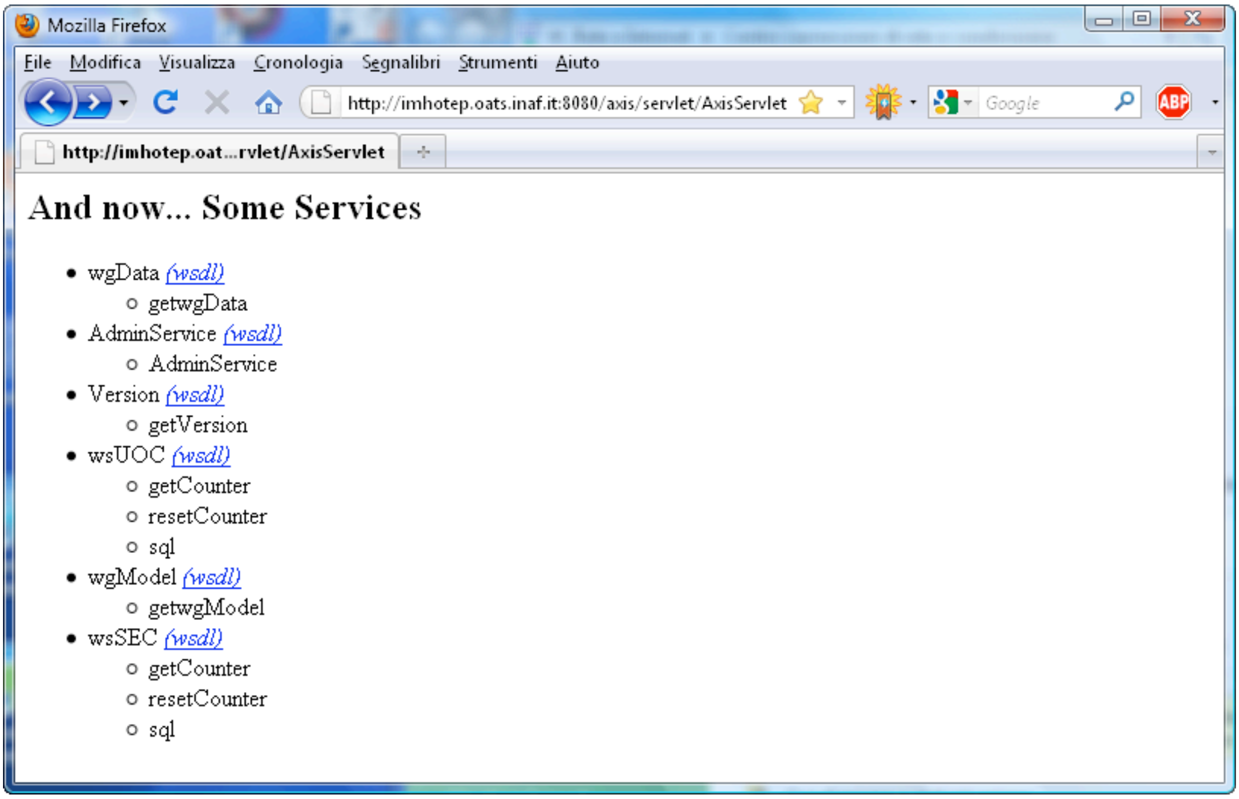
% java org.apache.axis.wsdl.WSDL2Java (WSDL-file-URL)

Applied to HEC it looks like:

% java org.apache.axis.wsdl.WSDL2Java http://imhotep.oats.inaf.it:8080/axis/services/wsSEC?wsdl

This will generate all java proxies classes needs to consume HEC methods.

In this release NetBeans is used as IDE but is possible to use Eclipse as well.


Mozilla Firefox
File Modifica Visualizza Cronologia Segnalibri Strumenti Aiuto
http://imhotep.oats.inaf.it:8080/axis/servlet/AxisServlet
Google
http://imhotep.oat...rvlet/AxisServlet

And now... Some Services

- wgData (wsdl)
  - getwgData
- AdminService (wsdl)
  - AdminService
- Version (wsdl)
  - getVersion
- wsUOC (wsdl)
  - getCounter
  - resetCounter
  - sql
- wgModel (wsdl)
  - getwgModel
- wsSEC (wsdl)
  - getCounter
  - resetCounter
  - sql

Figure 5 The wsSEC web service and the related wsdl file link

The related wsdl file is reported in the following.

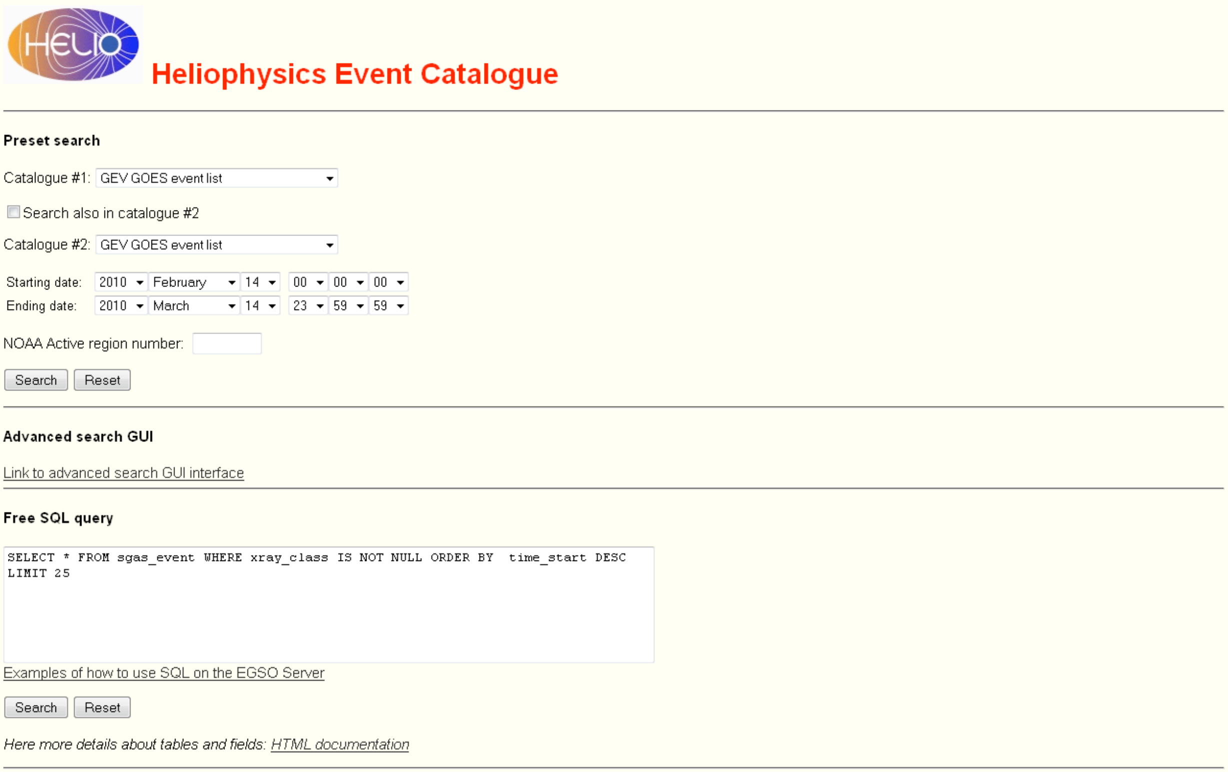
```
      – <wsdl:output name="getCounterResponse">
          <wsdlsoap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    – <wsdl:operation name="resetCounter">
        <wsdlsoap:operation soapAction=""/>
      – <wsdl:input name="resetCounterRequest">
          <wsdlsoap:body use="literal"/>
        </wsdl:input>
      – <wsdl:output name="resetCounterResponse">
          <wsdlsoap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    – <wsdl:operation name="sql">
        <wsdlsoap:operation soapAction=""/>
      – <wsdl:input name="sqlRequest">
          <wsdlsoap:body use="literal"/>
        </wsdl:input>
      – <wsdl:output name="sqlResponse">
          <wsdlsoap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  – <wsdl:service name="wsSECService">
    – <wsdl:port binding="impl:wsSECSoapBinding" name="wsSEC">
        <wsdlsoap:address location="http://imhotep.oats.inaf.it:8080/axis/services/wsSEC"/>
      </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

5 Graphical User Interface

The GUI provides the user with three different levels of search modes via a web page (**Figure 6**).

The outcome of all of them is an HTML page with the results formatted in tabular form, a VOTable and a text file.

In the following we illustrate such modes by means of some standard searches.



Heliophysics Event Catalogue

Preset search

Catalogue #1: GEV GOES event list

□ Search also in catalogue #2

Catalogue #2: GEV GOES event list

Starting date: 2010 February 14 00 00 00
Ending date: 2010 March 14 23 59 59

NOAA Active region number:

Advanced search GUI

Link to advanced search GUI interface

Free SQL query

```
SELECT * FROM sgas_event WHERE xray_class IS NOT NULL ORDER BY  time_start DESC
LIMIT 25
```

Examples of how to use SQL on the EGSO Server

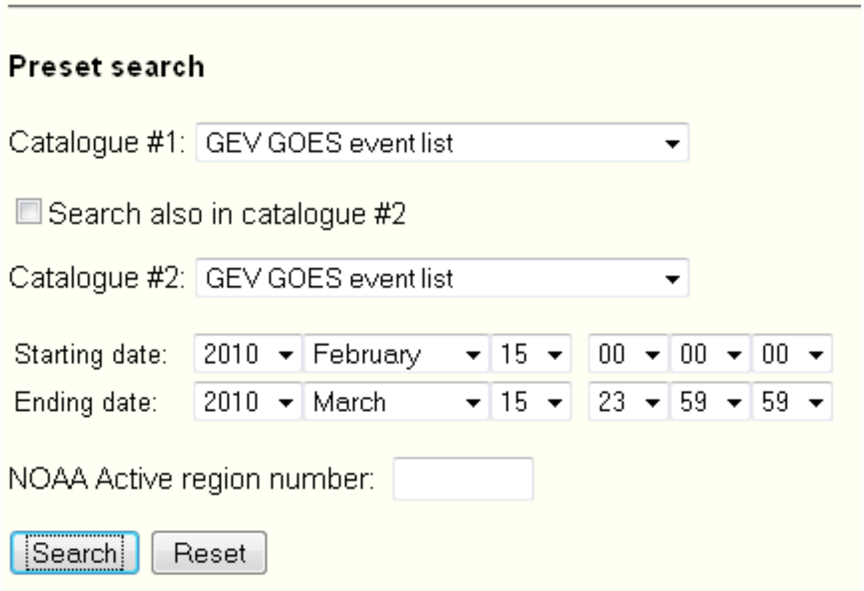
Here more details about tables and fields: HTML documentation

Figure 6 GUI main page

5.1 Search Mode via Presets

The Preset Search (**Figure 7**) is based on a coded set of options relevant to the available catalogues (**Figure 8**) and to the starting and ending date and time.

When the GEV GOES Event List is selected and the dates and times are specified, the relevant HTML (**Figure 10**), VOTable (**Figure 11**) and TXT file (**Figure 12**) are generated.



Preset search

Catalogue #1: GEV GOES event list

□ Search also in catalogue #2

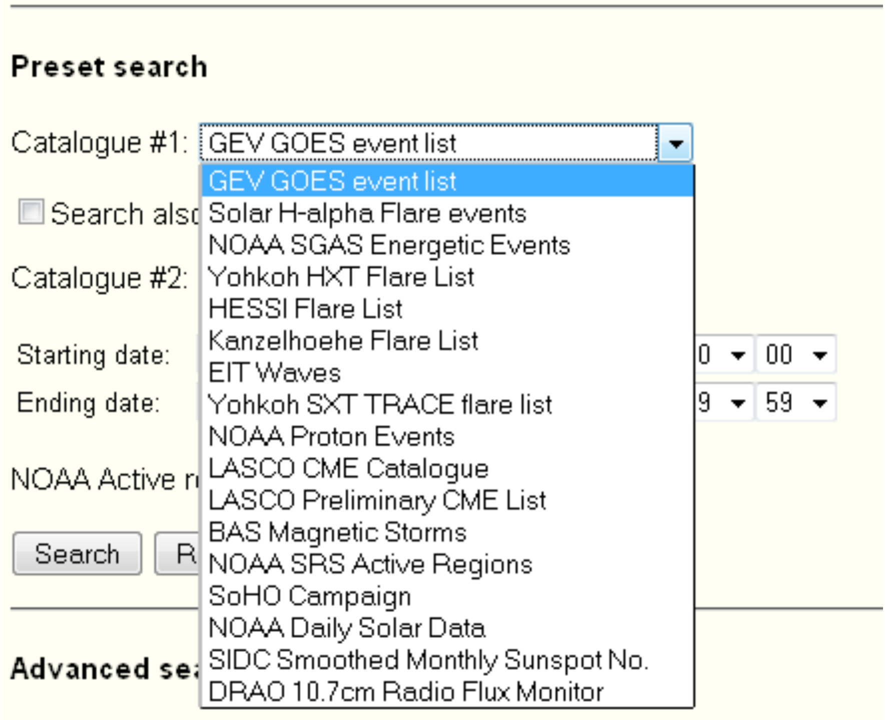
Catalogue #2: GEV GOES event list

Starting date: 2010 February 15 00 00 00

Ending date: 2010 March 15 23 59 59

NOAA Active region number:

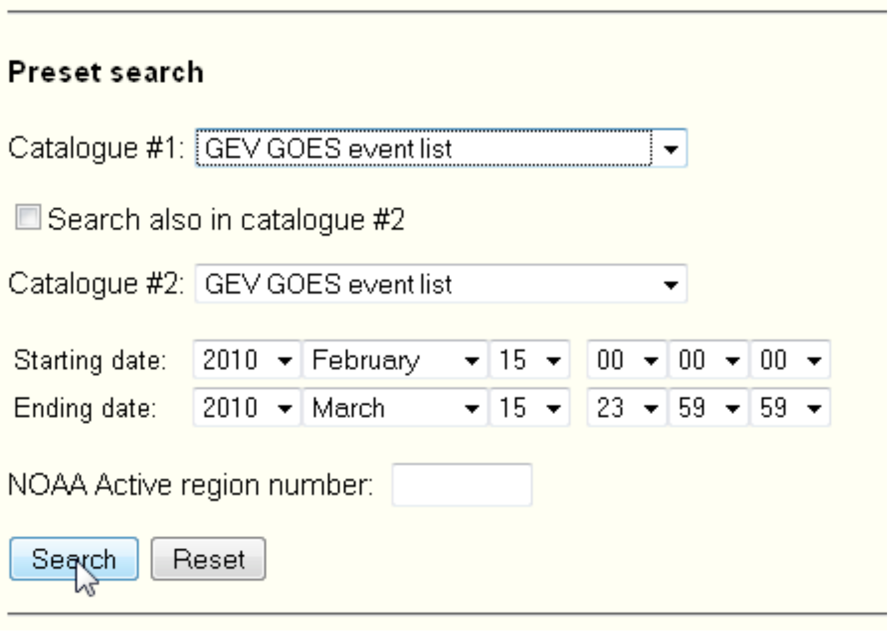
Figure 7 Search via presets



Preset search

Catalogue #1: GEV GOES event list

Figure 8 Catalogue selection in preset search



Preset search

Catalogue #1: GEV GOES event list

□ Search also in catalogue #2

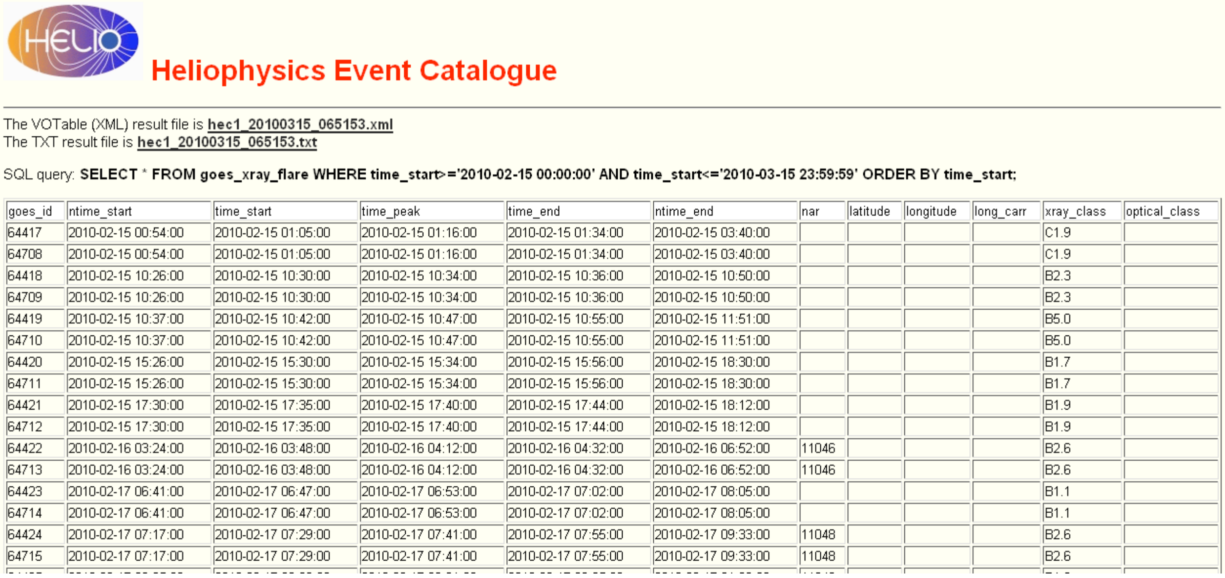
Catalogue #2: GEV GOES event list

Starting date: 2010 February 15 00 00 00

Ending date: 2010 March 15 23 59 59

NOAA Active region number:

Figure 9 Launch of a sample search for the GEV GOES event list



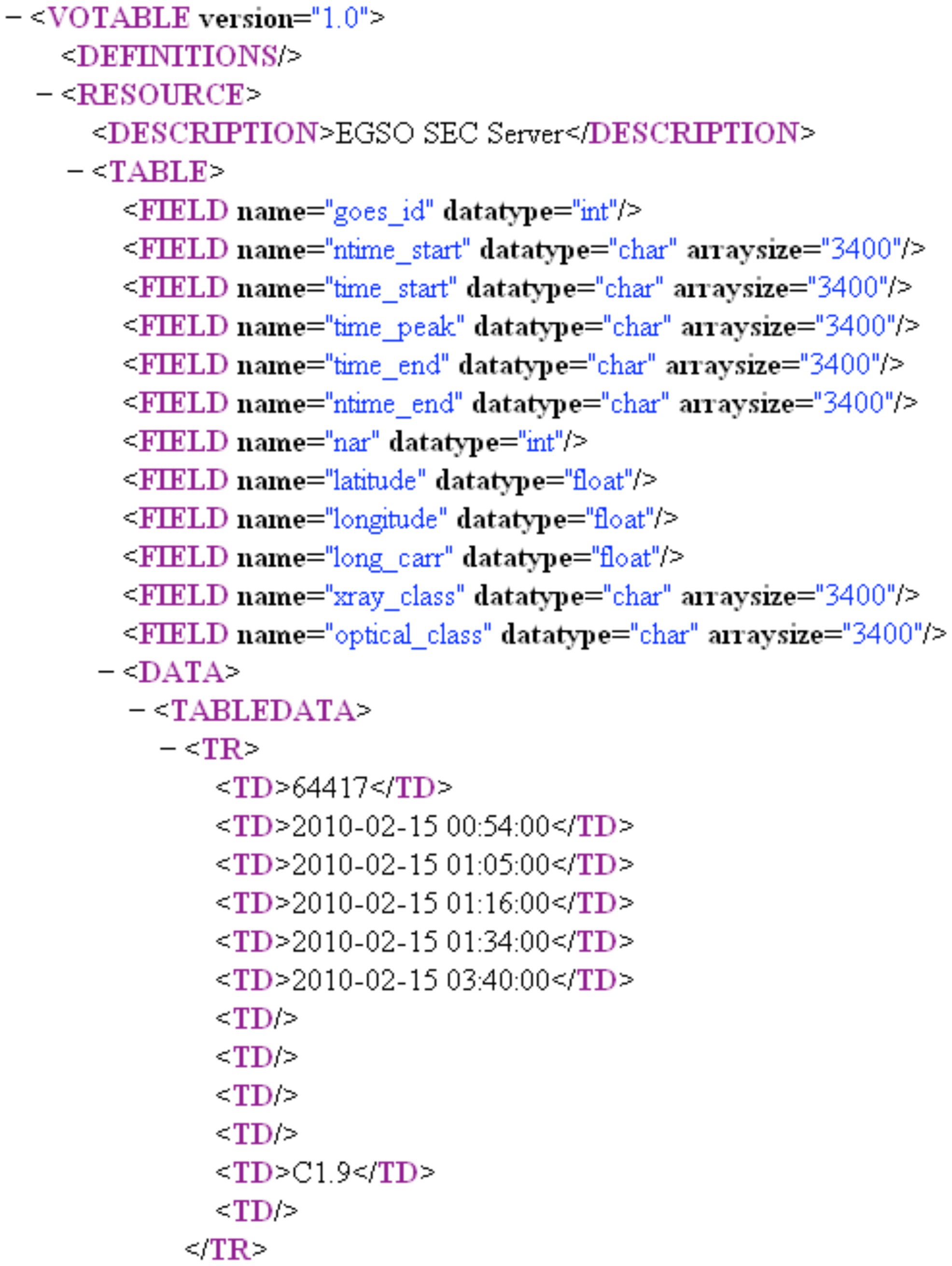
Heliophysics Event Catalogue

The VOTable (XML) result file is hec1_20100315_065153.xml
The TXT result file is hec1_20100315_065153.txt

SQL query: SELECT * FROM goes_xray_flare WHERE time_start>='2010-02-15 00:00:00' AND time_start<='2010-03-15 23:59:59' ORDER BY time_start;

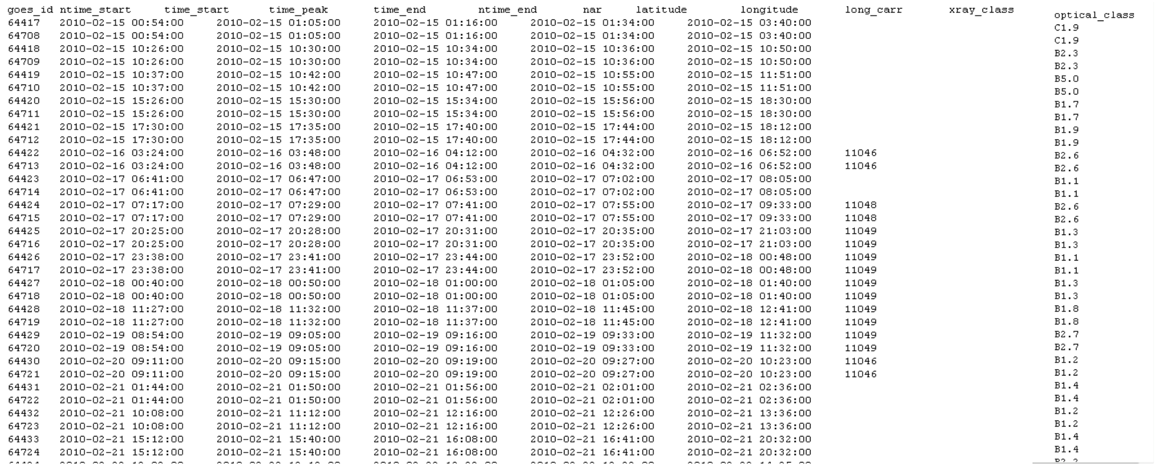
goes_id	ntime_start	time_start	time_peak	time_end	ntime_end	nar	latitude	longitude	long_carr	xray_class	optical_class
64417	2010-02-15 00:54:00	2010-02-15 01:05:00	2010-02-15 01:16:00	2010-02-15 01:34:00	2010-02-15 03:40:00					C1.9	
64708	2010-02-15 00:54:00	2010-02-15 01:05:00	2010-02-15 01:16:00	2010-02-15 01:34:00	2010-02-15 03:40:00					C1.9	
64418	2010-02-15 10:26:00	2010-02-15 10:30:00	2010-02-15 10:34:00	2010-02-15 10:36:00	2010-02-15 10:50:00					B2.3	
64709	2010-02-15 10:26:00	2010-02-15 10:30:00	2010-02-15 10:34:00	2010-02-15 10:36:00	2010-02-15 10:50:00					B2.3	
64419	2010-02-15 10:37:00	2010-02-15 10:42:00	2010-02-15 10:47:00	2010-02-15 10:55:00	2010-02-15 11:51:00					B5.0	
64710	2010-02-15 10:37:00	2010-02-15 10:42:00	2010-02-15 10:47:00	2010-02-15 10:55:00	2010-02-15 11:51:00					B5.0	
64420	2010-02-15 15:26:00	2010-02-15 15:30:00	2010-02-15 15:34:00	2010-02-15 15:56:00	2010-02-15 18:30:00					B1.7	
64711	2010-02-15 15:26:00	2010-02-15 15:30:00	2010-02-15 15:34:00	2010-02-15 15:56:00	2010-02-15 18:30:00					B1.7	
64421	2010-02-15 17:30:00	2010-02-15 17:35:00	2010-02-15 17:40:00	2010-02-15 17:44:00	2010-02-15 18:12:00					B1.9	
64712	2010-02-15 17:30:00	2010-02-15 17:35:00	2010-02-15 17:40:00	2010-02-15 17:44:00	2010-02-15 18:12:00					B1.9	
64422	2010-02-16 03:24:00	2010-02-16 03:48:00	2010-02-16 04:12:00	2010-02-16 04:32:00	2010-02-16 06:52:00	11046				B2.6	
64713	2010-02-16 03:24:00	2010-02-16 03:48:00	2010-02-16 04:12:00	2010-02-16 04:32:00	2010-02-16 06:52:00	11046				B2.6	
64423	2010-02-17 06:41:00	2010-02-17 06:47:00	2010-02-17 06:53:00	2010-02-17 07:02:00	2010-02-17 08:05:00					B1.1	
64714	2010-02-17 06:41:00	2010-02-17 06:47:00	2010-02-17 06:53:00	2010-02-17 07:02:00	2010-02-17 08:05:00					B1.1	
64424	2010-02-17 07:17:00	2010-02-17 07:29:00	2010-02-17 07:41:00	2010-02-17 07:55:00	2010-02-17 09:33:00	11048				B2.6	
64715	2010-02-17 07:17:00	2010-02-17 07:29:00	2010-02-17 07:41:00	2010-02-17 07:55:00	2010-02-17 09:33:00	11048				B2.6	

Figure 10 HTML page reporting the search results



```
– <VOTABLE version="1.0">
    <DEFINITIONS/>
  – <RESOURCE>
      <DESCRIPTION>EGSO SEC Server</DESCRIPTION>
    – <TABLE>
        <FIELD name="goes_id" datatype="int"/>
        <FIELD name="ntime_start" datatype="char" arraysize="3400"/>
        <FIELD name="time_start" datatype="char" arraysize="3400"/>
        <FIELD name="time_peak" datatype="char" arraysize="3400"/>
        <FIELD name="time_end" datatype="char" arraysize="3400"/>
        <FIELD name="ntime_end" datatype="char" arraysize="3400"/>
        <FIELD name="nar" datatype="int"/>
        <FIELD name="latitude" datatype="float"/>
        <FIELD name="longitude" datatype="float"/>
        <FIELD name="long_carr" datatype="float"/>
        <FIELD name="xray_class" datatype="char" arraysize="3400"/>
        <FIELD name="optical_class" datatype="char" arraysize="3400"/>
      – <DATA>
        – <TABLEDATA>
          – <TR>
              <TD>64417</TD>
              <TD>2010-02-15 00:54:00</TD>
              <TD>2010-02-15 01:05:00</TD>
              <TD>2010-02-15 01:16:00</TD>
              <TD>2010-02-15 01:34:00</TD>
              <TD>2010-02-15 03:40:00</TD>
              <TD/>
              <TD/>
              <TD/>
              <TD/>
              <TD>C1.9</TD>
              <TD/>
            </TR>
```

Figure 11 VOTable reporting the search results

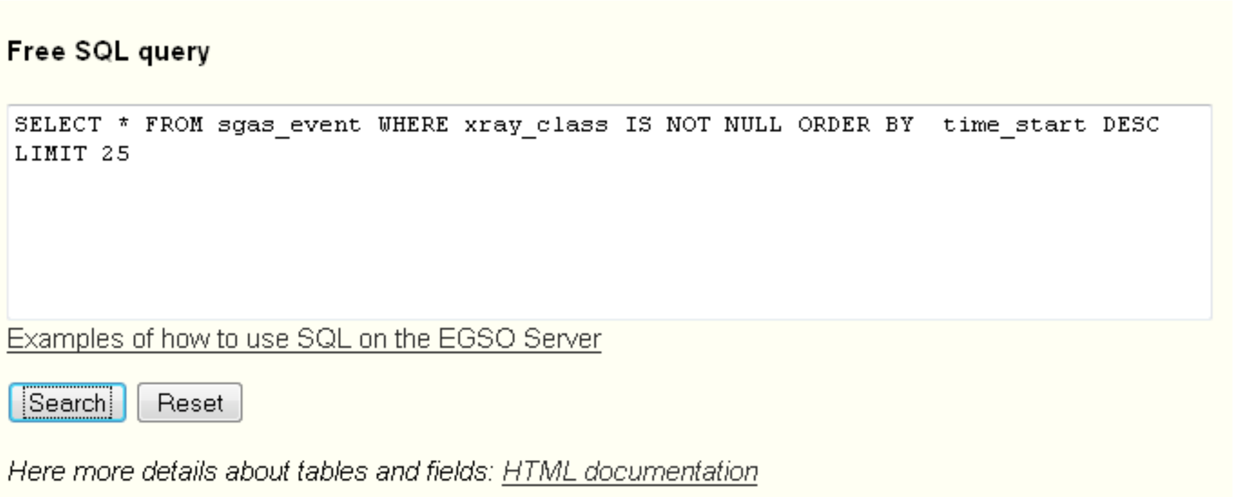


```
goes_id ntime_start     time_start      time_peak       time_end        ntime_end       nar     latitude        longitude       long_carr       xray_class      optical_class
64417   2010-02-15 00:54:00     2010-02-15 01:05:00     2010-02-15 01:16:00     2010-02-15 01:34:00     2010-02-15 03:40:00                     C1.9
64708   2010-02-15 00:54:00     2010-02-15 01:05:00     2010-02-15 01:16:00     2010-02-15 01:34:00     2010-02-15 03:40:00                     C1.9
64418   2010-02-15 10:26:00     2010-02-15 10:30:00     2010-02-15 10:34:00     2010-02-15 10:36:00     2010-02-15 10:50:00                     B2.3
64709   2010-02-15 10:26:00     2010-02-15 10:30:00     2010-02-15 10:34:00     2010-02-15 10:36:00     2010-02-15 10:50:00                     B2.3
64419   2010-02-15 10:37:00     2010-02-15 10:42:00     2010-02-15 10:47:00     2010-02-15 10:55:00     2010-02-15 11:51:00                     B5.0
64710   2010-02-15 10:37:00     2010-02-15 10:42:00     2010-02-15 10:47:00     2010-02-15 10:55:00     2010-02-15 11:51:00                     B5.0
64420   2010-02-15 15:26:00     2010-02-15 15:30:00     2010-02-15 15:34:00     2010-02-15 15:56:00     2010-02-15 18:30:00                     B1.7
64711   2010-02-15 15:26:00     2010-02-15 15:30:00     2010-02-15 15:34:00     2010-02-15 15:56:00     2010-02-15 18:30:00                     B1.7
64421   2010-02-15 17:30:00     2010-02-15 17:35:00     2010-02-15 17:40:00     2010-02-15 17:44:00     2010-02-15 18:12:00                     B1.9
64712   2010-02-15 17:30:00     2010-02-15 17:35:00     2010-02-15 17:40:00     2010-02-15 17:44:00     2010-02-15 18:12:00                     B1.9
64422   2010-02-16 03:24:00     2010-02-16 03:48:00     2010-02-16 04:12:00     2010-02-16 04:32:00     2010-02-16 06:52:00     11046           B2.6
64713   2010-02-16 03:24:00     2010-02-16 03:48:00     2010-02-16 04:12:00     2010-02-16 04:32:00     2010-02-16 06:52:00     11046           B2.6
64423   2010-02-17 06:41:00     2010-02-17 06:47:00     2010-02-17 06:53:00     2010-02-17 07:02:00     2010-02-17 08:05:00                     B1.1
64714   2010-02-17 06:41:00     2010-02-17 06:47:00     2010-02-17 06:53:00     2010-02-17 07:02:00     2010-02-17 08:05:00                     B1.1
64424   2010-02-17 07:17:00     2010-02-17 07:29:00     2010-02-17 07:41:00     2010-02-17 07:55:00     2010-02-17 09:33:00     11048           B2.6
64715   2010-02-17 07:17:00     2010-02-17 07:29:00     2010-02-17 07:41:00     2010-02-17 07:55:00     2010-02-17 09:33:00     11048           B2.6
64425   2010-02-17 20:25:00     2010-02-17 20:28:00     2010-02-17 20:31:00     2010-02-17 20:35:00     2010-02-17 21:03:00     11049           B1.3
64716   2010-02-17 20:25:00     2010-02-17 20:28:00     2010-02-17 20:31:00     2010-02-17 20:35:00     2010-02-17 21:03:00     11049           B1.3
64426   2010-02-17 23:38:00     2010-02-17 23:41:00     2010-02-17 23:44:00     2010-02-17 23:52:00     2010-02-18 00:48:00     11049           B1.1
64717   2010-02-17 23:38:00     2010-02-17 23:41:00     2010-02-17 23:44:00     2010-02-17 23:52:00     2010-02-18 00:48:00     11049           B1.1
64427   2010-02-18 00:40:00     2010-02-18 00:50:00     2010-02-18 01:00:00     2010-02-18 01:05:00     2010-02-18 01:40:00     11049           B1.3
64718   2010-02-18 00:40:00     2010-02-18 00:50:00     2010-02-18 01:00:00     2010-02-18 01:05:00     2010-02-18 01:40:00     11049           B1.3
64428   2010-02-18 11:27:00     2010-02-18 11:32:00     2010-02-18 11:37:00     2010-02-18 11:45:00     2010-02-18 12:41:00     11049           B1.8
64719   2010-02-18 11:27:00     2010-02-18 11:32:00     2010-02-18 11:37:00     2010-02-18 11:45:00     2010-02-18 12:41:00     11049           B1.8
64429   2010-02-19 08:54:00     2010-02-19 09:05:00     2010-02-19 09:16:00     2010-02-19 09:33:00     2010-02-19 11:32:00     11049           B2.7
64720   2010-02-19 08:54:00     2010-02-19 09:05:00     2010-02-19 09:16:00     2010-02-19 09:33:00     2010-02-19 11:32:00     11049           B2.7
64430   2010-02-20 09:11:00     2010-02-20 09:15:00     2010-02-20 09:19:00     2010-02-20 09:27:00     2010-02-20 10:23:00     11046           B1.2
64721   2010-02-20 09:11:00     2010-02-20 09:15:00     2010-02-20 09:19:00     2010-02-20 09:27:00     2010-02-20 10:23:00     11046           B1.2
64431   2010-02-21 01:44:00     2010-02-21 01:50:00     2010-02-21 01:56:00     2010-02-21 02:01:00     2010-02-21 02:36:00                     B1.4
64722   2010-02-21 01:44:00     2010-02-21 01:50:00     2010-02-21 01:56:00     2010-02-21 02:01:00     2010-02-21 02:36:00                     B1.4
64432   2010-02-21 10:08:00     2010-02-21 11:12:00     2010-02-21 12:16:00     2010-02-21 12:26:00     2010-02-21 13:36:00                     B1.2
64723   2010-02-21 10:08:00     2010-02-21 11:12:00     2010-02-21 12:16:00     2010-02-21 12:26:00     2010-02-21 13:36:00                     B1.2
64433   2010-02-21 15:12:00     2010-02-21 15:40:00     2010-02-21 16:08:00     2010-02-21 16:41:00     2010-02-21 20:32:00                     B1.4
64724   2010-02-21 15:12:00     2010-02-21 15:40:00     2010-02-21 16:08:00     2010-02-21 16:41:00     2010-02-21 20:32:00                     B1.4
```

Figure 12 Text file reporting the search results

5.2 SQL Free Search

For test purposes any SQL string can be manually entered in a window to be parsed and processed by the system (Figure 13).



Free SQL query

```
SELECT * FROM sgas_event WHERE xray_class IS NOT NULL ORDER BY  time_start DESC
LIMIT 25
```

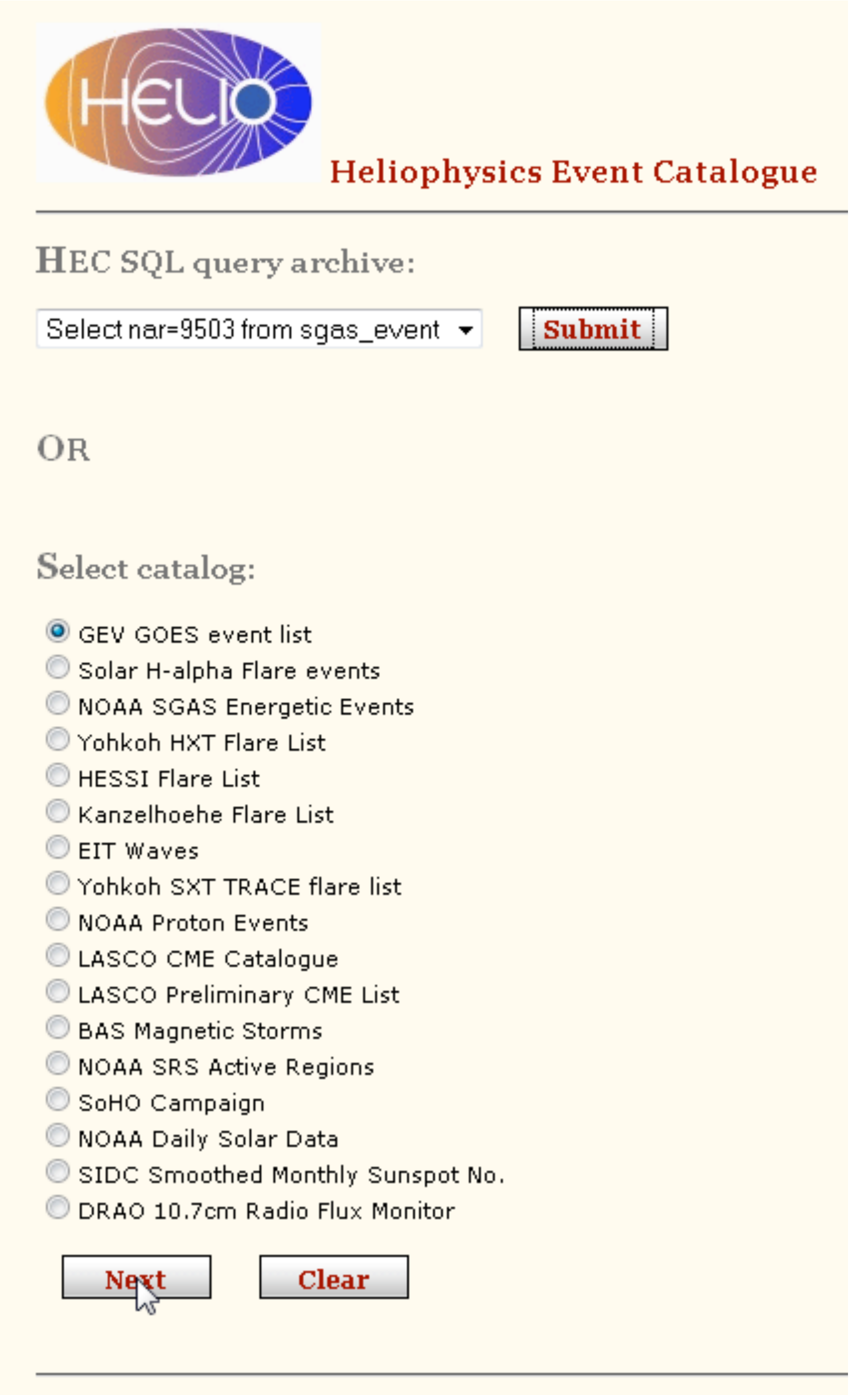
Here more details about tables and fields: HTML documentation

Figure 13 The free SQL query window

5.3 Advanced GUI

An advanced GUI has been developed for extended search tests on a single catalogue.

It can be used both via presets and via interactive selection of parameters (see **Figure 14**, **Figure 15**, **Figure 16**, **Figure 17**).



Heliophysics Event Catalogue

HEC SQL query archive:

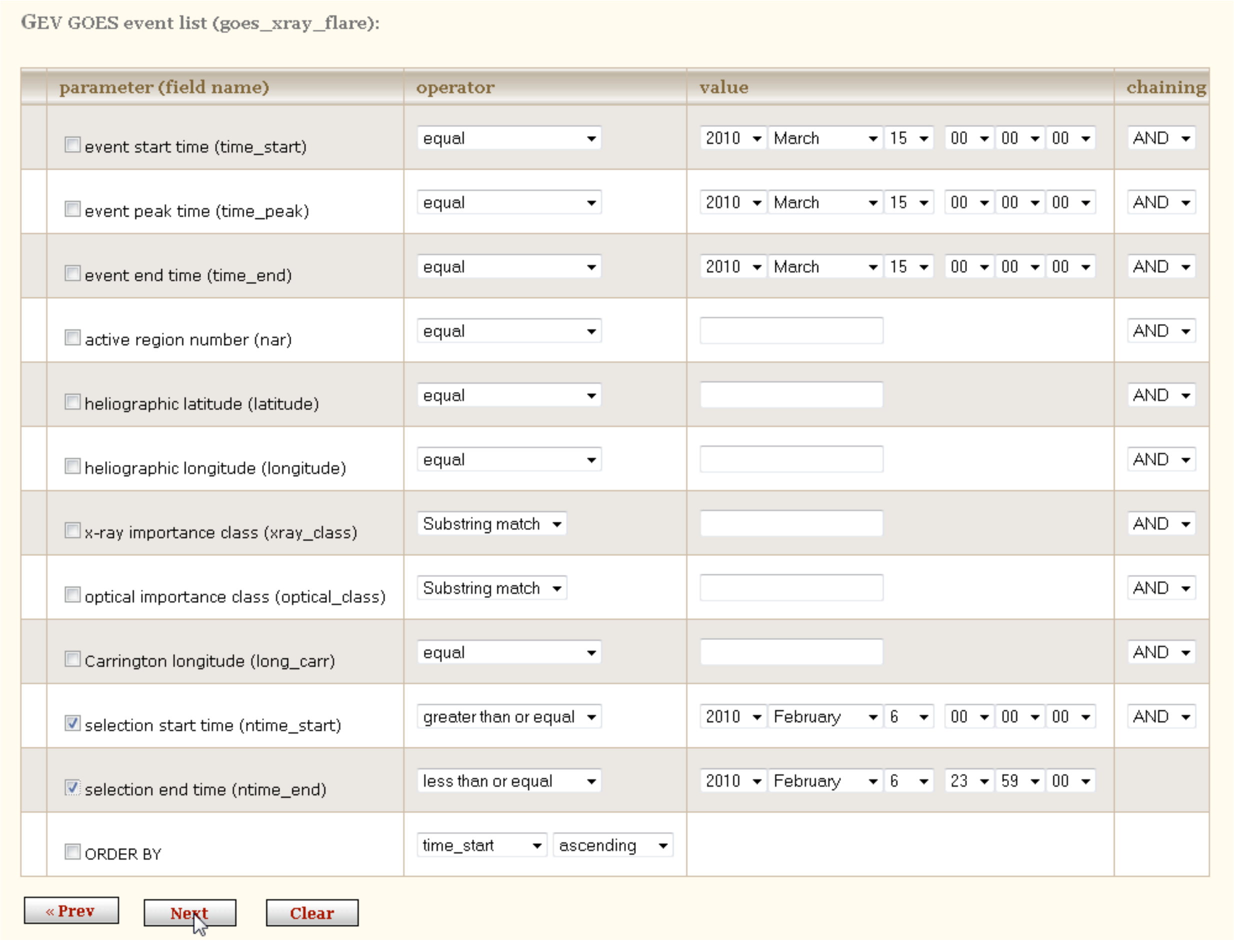
Select nar=9503 from sgas_event ▾

OR

Select catalog:

- ● GEV GOES event list
- ○ Solar H-alpha Flare events
- ○ NOAA SGAS Energetic Events
- ○ Yohkoh HXT Flare List
- ○ HESSI Flare List
- ○ Kanzelhoehe Flare List
- ○ EIT Waves
- ○ Yohkoh SXT TRACE flare list
- ○ NOAA Proton Events
- ○ LASCO CME Catalogue
- ○ LASCO Preliminary CME List
- ○ BAS Magnetic Storms
- ○ NOAA SRS Active Regions
- ○ SoHO Campaign
- ○ NOAA Daily Solar Data
- ○ SIDC Smoothed Monthly Sunspot No.
- ○ DRAO 10.7cm Radio Flux Monitor

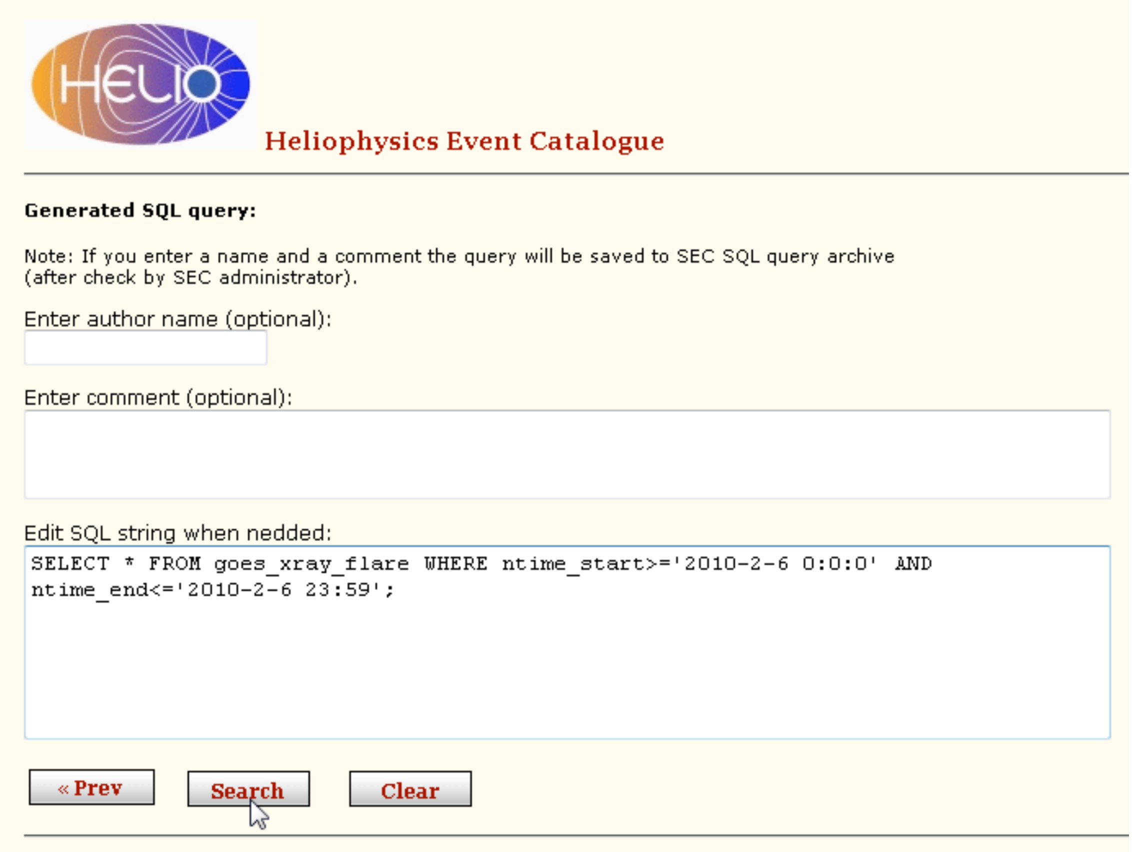
Figure 14 The main advanced GUI page



GEV GOES event list (goes_xray_flare):

parameter (field name)	operator	value	chaining
□ event start time (time_start)	equal	2010 March 15 00 00 00	AND
□ event peak time (time_peak)	equal	2010 March 15 00 00 00	AND
□ event end time (time_end)	equal	2010 March 15 00 00 00	AND
□ active region number (nar)	equal		AND
□ heliographic latitude (latitude)	equal		AND
□ heliographic longitude (longitude)	equal		AND
□ x-ray importance class (xray_class)	Substring match		AND
□ optical importance class (optical_class)	Substring match		AND
□ Carrington longitude (long_carr)	equal		AND
☑ selection start time (ntime_start)	greater than or equal	2010 February 6 00 00 00	AND
☑ selection end time (ntime_end)	less than or equal	2010 February 6 23 59 00	
□ ORDER BY	time_start ascending		

Figure 15 Conditions setup table for the advanced search



Heliophysics Event Catalogue

Generated SQL query:

Note: If you enter a name and a comment the query will be saved to SEC SQL query archive (after check by SEC administrator).

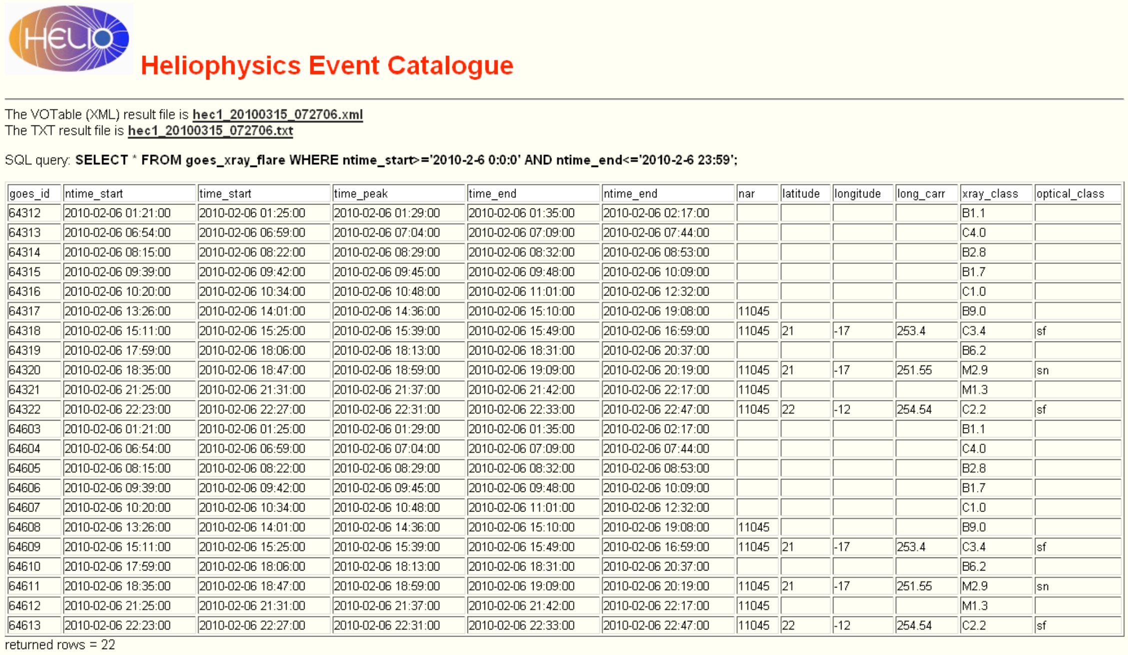
Enter author name (optional):

Enter comment (optional):

Edit SQL string when nedded:

```
SELECT * FROM goes_xray_flare WHERE ntime_start>='2010-2-6 0:0:0' AND
ntime_end<='2010-2-6 23:59';
```

Figure 16 SQL string built up from the selected advanced search conditions



Heliophysics Event Catalogue

The VOTable (XML) result file is **hec1_20100315_072706.xml**
The TXT result file is **hec1_20100315_072706.txt**

SQL query: **SELECT * FROM goes_xray_flare WHERE ntime_start>='2010-2-6 0:0:0' AND ntime_end<='2010-2-6 23:59';**

goes_id	ntime_start	time_start	time_peak	time_end	ntime_end	nar	latitude	longitude	long_carr	xray_class	optical_class
64312	2010-02-06 01:21:00	2010-02-06 01:25:00	2010-02-06 01:29:00	2010-02-06 01:35:00	2010-02-06 02:17:00					B1.1	
64313	2010-02-06 06:54:00	2010-02-06 06:59:00	2010-02-06 07:04:00	2010-02-06 07:09:00	2010-02-06 07:44:00					C4.0	
64314	2010-02-06 08:15:00	2010-02-06 08:22:00	2010-02-06 08:29:00	2010-02-06 08:32:00	2010-02-06 08:53:00					B2.8	
64315	2010-02-06 09:39:00	2010-02-06 09:42:00	2010-02-06 09:45:00	2010-02-06 09:48:00	2010-02-06 10:09:00					B1.7	
64316	2010-02-06 10:20:00	2010-02-06 10:34:00	2010-02-06 10:48:00	2010-02-06 11:01:00	2010-02-06 12:32:00					C1.0	
64317	2010-02-06 13:26:00	2010-02-06 14:01:00	2010-02-06 14:36:00	2010-02-06 15:10:00	2010-02-06 19:08:00	11045				B9.0	
64318	2010-02-06 15:11:00	2010-02-06 15:25:00	2010-02-06 15:39:00	2010-02-06 15:49:00	2010-02-06 16:59:00	11045	21	-17	253.4	C3.4	sf
64319	2010-02-06 17:59:00	2010-02-06 18:06:00	2010-02-06 18:13:00	2010-02-06 18:31:00	2010-02-06 20:37:00					B6.2	
64320	2010-02-06 18:35:00	2010-02-06 18:47:00	2010-02-06 18:59:00	2010-02-06 19:09:00	2010-02-06 20:19:00	11045	21	-17	251.55	M2.9	sn
64321	2010-02-06 21:25:00	2010-02-06 21:31:00	2010-02-06 21:37:00	2010-02-06 21:42:00	2010-02-06 22:17:00	11045				M1.3	
64322	2010-02-06 22:23:00	2010-02-06 22:27:00	2010-02-06 22:31:00	2010-02-06 22:33:00	2010-02-06 22:47:00	11045	22	-12	254.54	C2.2	sf
64603	2010-02-06 01:21:00	2010-02-06 01:25:00	2010-02-06 01:29:00	2010-02-06 01:35:00	2010-02-06 02:17:00					B1.1	
64604	2010-02-06 06:54:00	2010-02-06 06:59:00	2010-02-06 07:04:00	2010-02-06 07:09:00	2010-02-06 07:44:00					C4.0	
64605	2010-02-06 08:15:00	2010-02-06 08:22:00	2010-02-06 08:29:00	2010-02-06 08:32:00	2010-02-06 08:53:00					B2.8	
64606	2010-02-06 09:39:00	2010-02-06 09:42:00	2010-02-06 09:45:00	2010-02-06 09:48:00	2010-02-06 10:09:00					B1.7	
64607	2010-02-06 10:20:00	2010-02-06 10:34:00	2010-02-06 10:48:00	2010-02-06 11:01:00	2010-02-06 12:32:00					C1.0	
64608	2010-02-06 13:26:00	2010-02-06 14:01:00	2010-02-06 14:36:00	2010-02-06 15:10:00	2010-02-06 19:08:00	11045				B9.0	
64609	2010-02-06 15:11:00	2010-02-06 15:25:00	2010-02-06 15:39:00	2010-02-06 15:49:00	2010-02-06 16:59:00	11045	21	-17	253.4	C3.4	sf
64610	2010-02-06 17:59:00	2010-02-06 18:06:00	2010-02-06 18:13:00	2010-02-06 18:31:00	2010-02-06 20:37:00					B6.2	
64611	2010-02-06 18:35:00	2010-02-06 18:47:00	2010-02-06 18:59:00	2010-02-06 19:09:00	2010-02-06 20:19:00	11045	21	-17	251.55	M2.9	sn
64612	2010-02-06 21:25:00	2010-02-06 21:31:00	2010-02-06 21:37:00	2010-02-06 21:42:00	2010-02-06 22:17:00	11045				M1.3	
64613	2010-02-06 22:23:00	2010-02-06 22:27:00	2010-02-06 22:31:00	2010-02-06 22:33:00	2010-02-06 22:47:00	11045	22	-12	254.54	C2.2	sf

returned rows = 22

Figure 17 Advanced search results