**Department of Computer Science**
**Georgia State University**
**Atlanta, GA, United States.**

# Designing a Storage Efficient and Faster Heliophysics Events Knowledgebase (HEK)

Andre Kenneth Chase Randall[1]; Ahmet Kucuk[2]; Shah Muhammad Hamdi[2]; Soukaina Filali[2]
1. Honor College, Department of Computer Science (Undergradeuate)
2. Department of Computer Science (Graduate)

# Table of Contents

## 1. Introduction

Dr. Rafal Angryk, Associate Professor and Director of DMLab at Georgia State University, gave a research assignment. Our research assignment not only covered the study materials of the course CSC 4710 (cross listed as 6710): Honors Database Systems, but also covered an ongoing effort of DMLab in making an efficient Heliophysics Events Knowledgebase "HEK". Owned by Lockheed Martin, Inc., the HEK contained solar image data captured by NASA's Solar Data Observatory (SDO). SDO sent the captured image to different modules that extracted features from those events. HEK stored those modules event data as an event type database. So far, solar physicists divided the solar image data into 22 solar event types stored in the HEK. The operation of HEK in storing solar events in explained in Fig. 1.

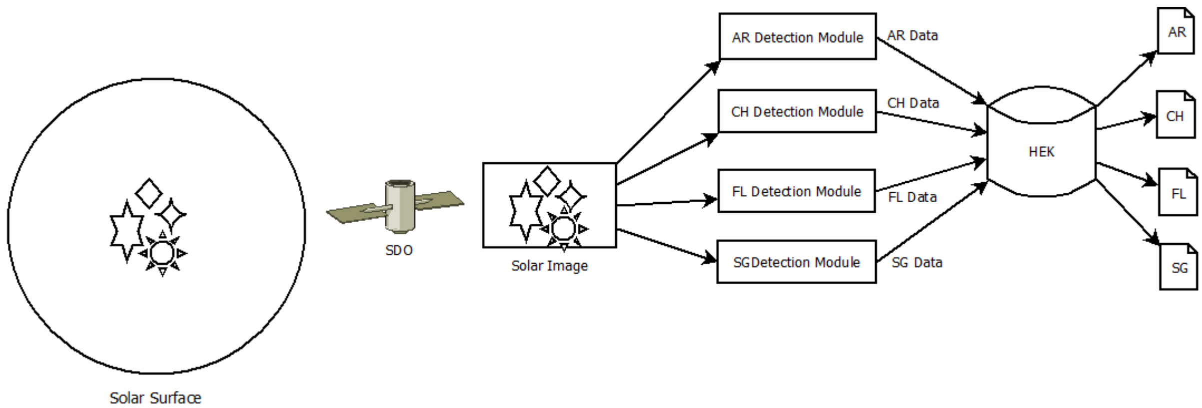


Fig. 1 HEK stores four solar event types in four data files after they are captured by SDO

Recently, DMLab of GSU posed the question of how to enhance the HEK performance with a new design. The new design should encompass features such as, handling spatial, temporal and spatiotemporal queries; storing trajectories of events; having a cache of recently stored events; and predicting future solar events.

## 2. Problem Definition:

Fig. 1 shows that the data stored in HEK suffers from a lot of redundancy. When a picture is captured by SDO, it contains information (e.g. observation instrument) that is the same for all the events present in the picture. As HEK stores event information in separate files, same information is redundantly inserted, which wastes a lot of storage. With the concepts of designing relational databases and some domain knowledge, is it possible to redesign HEK so that it can be more storage efficient and faster in handling queries? Moreover, does the data mining research efforts make it possible to track solar events and predict future events? As trajectory data and future events data are available, can they be included in our database? Also, who to improve caching for fast retrieval of recently stored event types. So, the problem statement objectives becomes the following:

- To redesign HEK to enhance performance.
- To handle spatial, temporal and spatiotemporal queries.
- To store trajectories of event instances.
- To cache recently recorded events instances.
- To allocate predicted future event instances.

## 3. Contextual Data Flow Diagram

Our database users following in the following categories: solar physicists; computer scientist (big data and data mining researchers); and the general enthusiasts (see diagram below).

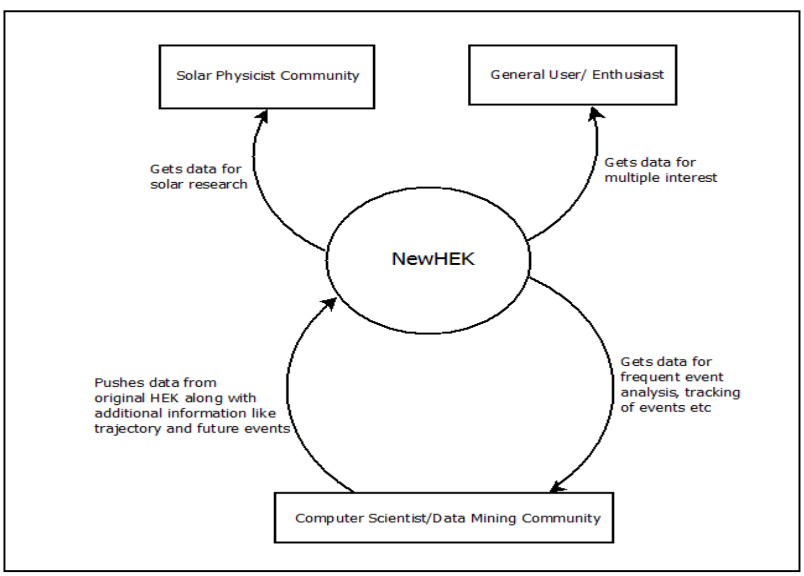


Fig. 2 Data Flow Diagram

## 4. System Requirements:

As solar event data holds both spatial and temporal information, the system requirements are divided into four types – spatial query requirements, temporal query requirements, spatiotemporal query requirements and non-spatiotemporal query requirements. Examples of query types follows:

i. Spatial query:
   a. A point in solar surface is given.
      Query: Is there an event instance at that point?
   b. Locations of two event instances are given.
      Query: What is the distance between those two?
   c. Location of an event instance is given.
      Query: Is their intersection of event instances? What are they?
   d. Spatial kNN queries.
      Query: How many event instances are there in a given neighborhood?

ii. Temporal query:
      Query for event instances before/after/interval of a given time

iii. Spatiotemporal query:
      An event type, start time and end time is given. What is the spatial coordinates of the event instances occurring at that time?

iv. Non-spatiotemporal query:
      Query for the attributes that are neither spatial nor temporal, e.g. observatory name, observation wave length, observation instrument, Feature Recognition Method (FRM) name etc.

## 5. Data Analysis

Our lack of knowledge, that is, data analysis, became the most challenging research aspect. Database design requires data type domain knowledge. Given our computer science background, which primarily focused on programming language concepts, versus an Astrophysics background, which primarily focused on interpretation of astronomical observations, we spent a significant amount of time understanding solar event types and their attributes.

### 5.1. Downloading the data

We downloaded data from HEK using a Java program [] named *QueryHEK*. *QueryHEK* downloads HEK data in XML or JSON format as per user's need. Another Java program *JsonParser* parses the downloaded XML or JSON files into tab delimited text files. Data of four event types – Active Region (AR), Coronal Hole (CH), Flare (FL) and Sigmoid (SG) were downloaded. The downloaded data consisted of all the event instances of these four types in the time range of January 1, 2012 to December 31, 2014.

### 5.2 Analyzing the data

We used some Java and R programs to better investigate the nature of each attribute. We summarize our findings as follows:

- Three data types attributes are there: Boolean, String and Numeric.
- Many attributes hold only null values. The representation of null values is not consistent. Some attributes hold actual *null* value, some hold the string "null" and some are just an empty field value.
- In all four files, the common key used is *kb_archivid*. The attribute kb_archivid stores information about the event type, event generation time and event recording time. For example, we show a value of the attribute kb_archivid in the "FL" table. ivo://helio-informatics.org/FL_FlareDetectiveTriggerModule_20120103_221710_2012-01-01T02:23:26.080_1
- In the original HEK files, we found 169 attributes for AR, 128 for CH, 147 for FL and 135 for SG.
- We found 120 attributes are common in all four files.
- Then we found 49 special attributes for AR (they are present only in the "AR" file, not in CH, FL or SG), 8 for CH, 27 for FL and 15 for SG.

## 6. Design of the Database

We designed the database in Extended Entity Relationship (EER) model. As we found 120 common attributes among all four event types, we made a generalized entity type and named it *Generic_event*. Then we made four specialized entity types for each event type and allocated the special attributes with them. We made the relationship between super-class and subclass as partial disjoint. It is partial because, we have designed only four event types so far. As there are total 22 event types, we are not imposing total participation. It is disjoint, because event types do not have shared identity, i.e. if an event instance can never be of two different types. *Generic_event* is related to the entity type *Trajectory* by the relationship *forms*. For storing future events, we made another entity type *future_event.*

Our design implies that there must be two INSERT operations for each occurrence of event instance. Suppose an event instance of AR is found. Then there will be two INSERTs, one at the Generic_event table and another at the AR table.

The EER model, relational model and a database instance with respect to some trajectory of events are as follows.

### 6.1. EER Model

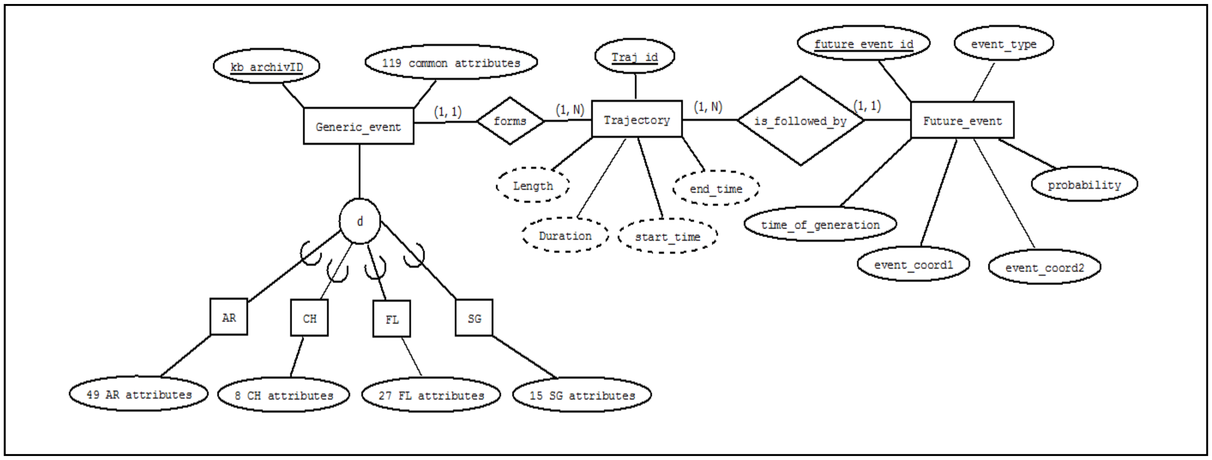


Fig. 3 EER Model

### 6.2. Relational Model

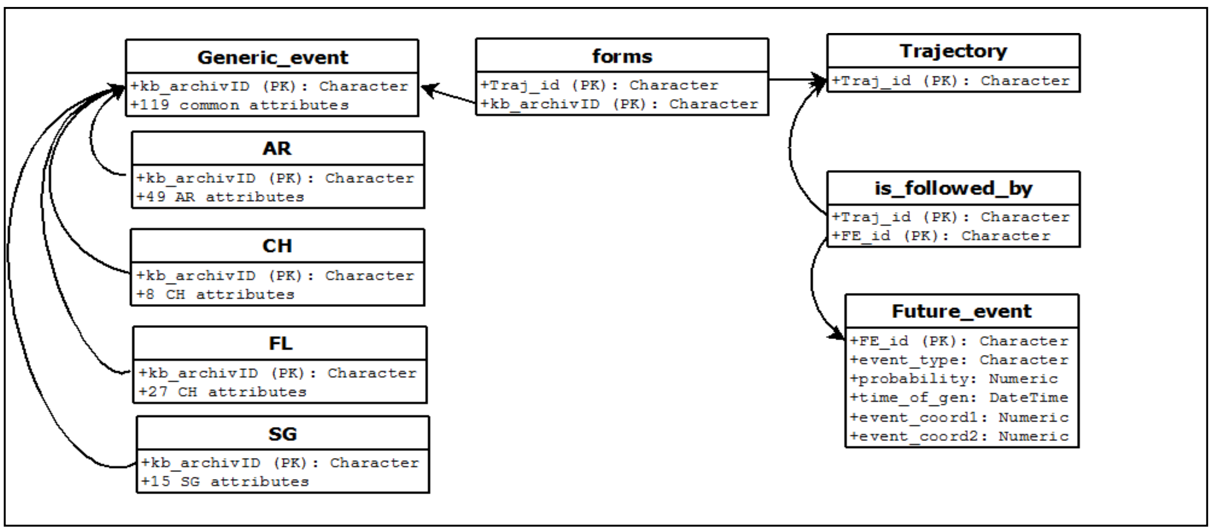


Fig. 4 Relational Model

### Database Instance with respect to Trajectories of Event Instances

Suppose three event instances of SG and two event instances of CH are recorded in the table *Generic_event* and corresponding tables SG and CH. Tracking algorithm says that the three instances of SG forms one trajectory and two instances of CH forms another trajectory. After analyzing the trajectories, two instances of SG and one instance of CH are predicted as future event instances. In the following figures, we show how these information are stored in our database.

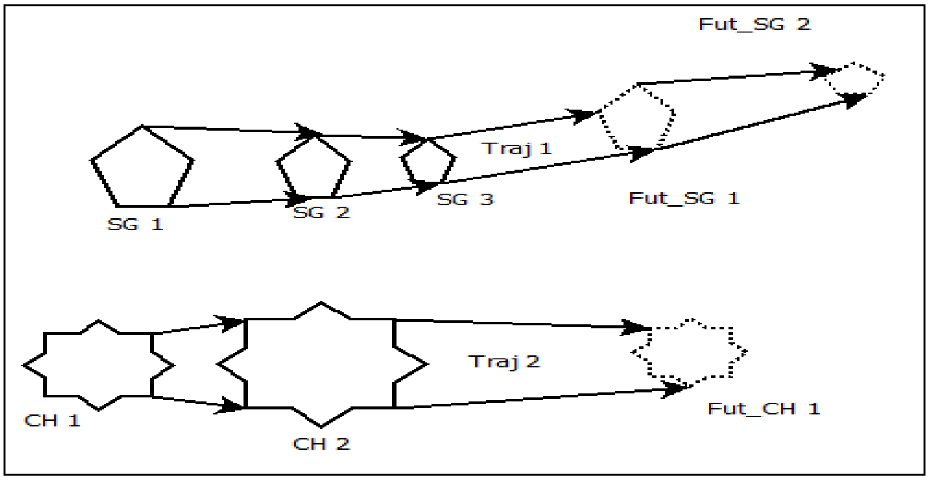


Fig. 5 Two trajectories of three instances of SG and two instances of CH; two future SG instances and one future CH instance are predicted from the trajectories.

Table name: is_followed_by

| Traj_id | FE_id |
|---|---|
| Traj_1 | Fut_SG_1 |
| Traj_1 | Fut_SG_2 |
| Traj_2 | Fut_CH_1 |

Table name: Future_event

| FE_id | Event_type | Probability | Time_of_generation | All spatial attributes |
|---|---|---|---|---|
| Fut_SG_1 | SG | 0.7 | … | … |
| Fut_SG_2 | SG | 0.68 | … | … |
| Fut_CH_1 | CH | 0.5 | … | … |

Table name: forms

| Traj_id | kb_archivID |
|---|---|
| Traj_1 | SG_1 |
| Traj_1 | SG_2 |
| Traj_1 | SG_3 |
| Traj_2 | CH_1 |
| Traj_2 | CH_2 |

Table name: Trajectory

| Traj_id |
|---|
| Traj_1 |
| Traj_2 |

Table name: Generic_event

| kb_archivID | 119 common attributes |
|---|---|
| SG_1 | … |
| SG_2 | … |
| SG_3 | … |
| CH_1 | … |
| CH_2 | … |

Table name: SG

| kb_archivID | 15 SG attributes |
|---|---|
| SG_1 | … |
| SG_2 | … |
| SG_3 | … |

Table name: CH

| kb_archivID | 8 CH attributes |
|---|---|
| CH_1 | … |
| CH_2 | … |

Fig. 6 Database instance for storing those trajectories

## 7. Data Dictionary

In the data dictionary, we enlist every attribute in our database with their data types and definition.

| Attribute Name | Type | Definition |
|---|---|---|
| Event_Probability | float | Probability or Confidence Level that event occured (bet. 0 and 1) |
| Event_Importance | float | Rating or importance of the event (between 0 and 1). Can be used by automated methods to denote a metric. |
| Event_Type | string | Event Type (e.g. 'FL: Flare' or 'AR: ActiveRegion') |
| KB_ArchivDate | string | Date when VOEvent entry was imported into Knowledge Base |
| KB_ArchivID | string | Unique internal ID of VOEvent entry |
| KB_Archivist | string | Name of Archivist (internal. user should leave blank) |
| KB_ArchivURL | string | URL of VOEvent entry (internal. user should leave blank) |
| Event_CoordSys | string | Coordinate system type (Choose between UTC-HGS-TOPO [Heliographics Stonyhurst]; UTC-HPR-TOPO [Helioprojective]; UTC-HGC-TOPO[Heliographic Carrington]; UTC-HCR-TOPO[Helio-centric radial]) |
| Event_CoordUnit | string | Units of coordinates (e.g. "deg, deg" for UTC-HGS-TOP0) |
| Event_EndTime | string | Time when event ends (e.g. 2004-02-14T02:00:01) |
| Event_StartTime | string | Time when event starts (e.g. 2004-02-14T02:00:01) |
| Event_Expires | string | Useful for reporting events before they are complete (e.g. 2004-02-14T02:00:01) |
| Event_Coord1 | float | Coordinate 1 of mean location of event |
| Event_Coord2 | float | Coordinate 2 of mean location of event |
| Event_Coord3 | float | Coordinate 3 of mean location of event (optional. Suitable for use with STEREO SECCHI events) |
| Event_MapURL | string | URL to an image/intensity map |
| Event_MaskURL | string | URL to files which contain masks (e.g. binary masks) of region of interest. |
| Event_PeakTime | string | Peak time of a flare (e.g. '2003-02-12T23:03:01') |
| Event_C1Error | float | Uncertainty in Coord1 of the mean location of the event. |
| Event_C2Error | float | Uncertainty in Coord2 of the mean location of the event. |
| Event_ClippedSpatial | string | Whether the spatial extent of the event is wholly contained within the data set ('T' or 'F'). |
| Event_ClippedTemporal | string | Whether the temporal duration of the event is wholly contained within the data set ('T' or 'F'). |
| Event_TestFlag | string | A boolean flag to indicate that the event is for testing purposes ('T' or 'F') |
| Event_Description | string | Description of the event |
| FRM_Contact | string | Contact information of Feature Recognition Method (FRM) |

| | | |
|---|---|---|
| FRM_DateRun | string | Date when Feature Recognition Method (FRM) was run (e.g. 2004-02-15T02:00:01) |
| FRM_HumanFlag | string | Whether a Human identified the event ("T" or "F") |
| FRM_Identifier | string | Username for Knowledge Base |
| FRM_Institute | string | Institute where the Feature Recognition Method (FRM) orginates |
| FRM_Name | string | Name of Feature Recognition Method (e.g. "Mark Cheung" or CACTUS") |
| FRM_ParamSet | string | Values of parameters (e.g. "threshold=0.1") |
| FRM_VersionNumber | float | Version number of automated Feature Recognition Method (Put age if Human. Just kidding. In this case put 1.0) |
| FRM_URL | string | URL to webpage containing information about the Feature Recognition Method |
| FRM_SpecificID | string | The specific ID of this event/feature assigned by the Feature Recognition Method |
| OBS_Observatory | string | Name of Observatory (e.g. SOHO) |
| OBS_ChannelID | string | Name of Channel of the instrument (e.g. "G band") |
| OBS_Instrument | string | Name of Instrument (e.g. "SOT") |
| OBS_MeanWavel | float | Mean wavelength (preferably in Angstroms) |
| OBS_WavelUnit | string | Unit of OBS_MeanWavel (preferably "Angstroms") |
| OBS_Title | string | Observational title |
| Bound_CCNsteps | long | Number of steps in bounding chain code (useful for coronal hole boundaries) |
| Bound_CCStartC1 | float | Beginning Coord1 of chain code |
| Bound_CCStartC2 | float | Beginning Coord2 of chain code |
| Bound_ChainCode | string | List of vertices for polygon (ordered list delimited by commas. e.g. 'x1, y1, x2, y2, x3, y3, x1, y1') |
| BoundBox_C1LL | float | Coord1 of lower-left corner of bounding box |
| BoundBox_C2LL | float | Coord2 of lower-left corner of bounding box |
| BoundBox_C1UR | float | Coord1 of upper-right corner of bounding box |
| BoundBox_C2UR | float | Coord2 of upper-right corner of bounding box |
| ChainCodeType | string | Type of chain code (Use "ordered list") |
| RasterScan | string | Field for ascii string of raster scan |
| RasterScanType | string | Type of raster scan (E.g. "EGSO_SFC" if you are using the EGSO Solar Feature Catalogue convection for the raster scan) |
| Skel_ChainCode | string | Skeleton code. A skeleton code is like a chain code except it isn't closed. (ordered list delimited by commas. e.g. 'x1, y1, x2, y2, x3, y3') |
| Skel_Curvature | float | Curvature of skeleton |
| Skel_Nsteps | integer | Number of steps in skeleton |
| Skel_StartC1 | float | Beginning Coord 1 of skeleton |
| Skel_StartC2 | float | Beginning Coord 2 of skeleton |

| | | |
|---|---|---|
| AR_McIntoshCls | string | Active Region McIntosh class |
| AR_MtWilsonCls | string | Active Region Mt Wilson class |
| AR_ZurichCls | string | Active Region Zurich class |
| AR_PenumbraCls | string | Active Region Penumbra class |
| AR_CompactnessCls | string | Active Region Compactness class |
| AR_NOAAclass | string | Active Regon NOAA class |
| AR_NOAANum | long | NOAA designated Active Region Number (e.g. 10930) |
| AR_NumSpots | integer | Number of spots in Active region |
| AR_Polarity | integer | Polarity of Active region ('1' or '-1' for positive and negative respectively) |
| AR_SpotAreaRaw | float | Area of spots in active region in plane of sky |
| AR_SpotAreaRawUncert | float | Uncertainty of AR_SpotAreaRaw |
| AR_SpotAreaRawUnit | string | Units of AR_SpotAreaRaw |
| AR_SpotAreaRepr | float | Reprojected area of spots in heliographic units |
| AR_SpotAreaReprUncert | float | Uncertainty Reprojected area of spots in heliographic units |
| AR_SpotAreaReprUnit | string | Units of AR_SpotAreaReprUnit (e.g. 'millihemispheres' or 'steradians') |
| SHARP_NOAA_ARS | string | Comma separated list of NOAA_ARS within lat/lon bounding box. Can include ones without NOAA# at time of event. |
| IntensMin | float | Minimum intensity of pixels |
| IntensMax | float | Maximum intensity of pixels |
| IntensMean | float | Mean intensity of pixels |
| IntensMedian | float | Mdian intensity of pixels |
| IntensVar | float | Variance of intensity of pixels |
| IntensSkew | float | Skewness of intensity of pixels |
| IntensKurt | float | Kurtosis of intensity of pixels |
| IntensTotal | float | Sum of intensities of pixels |
| IntensUnit | string | Units of intensity |
| FL_GOESCls | string | GOES Flare class (e.g. 'X11') |
| CME_RadialLinVel | float | Radial Linear fit radial velocity of CME |
| CME_RadialLinVelUncert | float | Uncertainty in CME_RadialLinVel |
| CME_RadialLinVelMin | float | Mininum linear radial velocity of CME |
| CME_RadialLinVelMax | float | Maximum linear radial velocity of CME |
| CME_RadialLinVelStddev | float | Standard Deviation of radial velocity of CME |
| CME_RadialLinVelUnit | string | Units for Radial velocity of CME (e.g. 'km/s') |
| CME_AngularWidth | float | Angular width of CME |
| CME_AngularWidthUnit | string | Units for angular width of CME (e.g. 'deg') |
| CME_Accel | float | Acceleration of CME |
| CME_AccelUncert | float | Uncertainty of CME acceleration |
| CME_AccelUnit | string | Units for CME acceleration (e.g. 'km/s/s') |
| CME_Mass | float | Mass contained in CME (e.g. '1e17') |
| CME_MassUncert | float | Uncertainty in mass contained in cme |

| | | |
|---|---|---|
| CME_MassUnit | string | Units for CME mass (e.g. 'g') |
| Area_AtDiskCenter | float | Area of event at disk center |
| Area_AtDiskCenterUncert | float | Uncertainty of area at disk center |
| Area_Raw | float | Area of event in sky plane |
| Area_Uncert | float | Uncertainty of area in sky plane |
| Area_Unit | string | Units of area in sky plane (e.g. 'arcsec2') |
| Event_Npixels | long | Number of pixels pertaining to event |
| Event_PixelUnit | string | Units of values given in pixels |
| FreqMaxRange | float | Maximum of the frequency range of oscillation |
| FreqMinRange | float | Minimum of the frequency range of oscillation |
| FreqPeakPower | float | Frequency at which power spectrum peaks |
| FreqUnit | string | Units of frequency (e.g. 'Hz') |
| IntensMaxAmpl | float | Maximum amplitude of oscillation in intensity signal |
| IntensMinAmpl | float | Minimum amplitude of oscillation in intensity signal |
| OscillNPeriods | float | Number of periods detected in oscillation |
| OscillNPeriodsUncert | float | Uncertainty of number of periods detected in oscillation |
| PeakPower | float | Peak power of oscillation |
| PeakPowerUnit | string | Units of peak power of oscillation |
| VelocMaxAmpl | float | Maximum amplitude of oscillation in velocity signal (e.g. doppler signal) |
| VelocMaxPower | float | Maximum power of oscillation in velocity signal |
| VelocMaxPowerUncert | float | Uncertainty in max power in velocity signal |
| VelocMinAmpl | float | Minimum amplitude in oscillating velocity signal |
| VelocUnit | string | Units for velocity (e.g. 'km/s') |
| WaveDisplMaxAmpl | float | Maximum amplitude of oscillation in displacement signal |
| WaveDisplMinAmpl | float | Minimum amplitude of oscillatoin in displacement signal |
| WaveDisplUnit | string | Units for displacement amplitude (e.g. 'arcsec') |
| WavelMaxPower | float | Wavelength at which spatial power spectrum peaks |
| WavelMaxPowerUncert | float | Uncertainty of WavelMaxPower |
| WavelMaxRange | float | Maximum wavelength of wavelength range for spatial oscillation |
| WavelMinRange | float | Minimum wavelength of wavelength range for spatial oscillation |
| WavelUnit | string | Units for spatial oscillation wavelength (e.g. 'km') |
| EF_PosPeakFluxOnsetRate | float | Emergence rate of positive polarity flux |
| EF_NegPeakFluxOnsetRate | float | Emergence rate of negative polarity flux |
| EF_OnsetRateUnit | string | Emergence rate unit (e.g. 'Mx/s') |
| EF_SumPosSignedFlux | float | Total positive signed flux at completion |
| EF_SumNegSignedFlux | float | Total negative signed flux at completion (negative number) |
| EF_FluxUnit | string | Flux unit (e.g. 'Mx') |
| EF_AxisOrientation | float | Axis orientation of emerging flux pair (CCW from parallels in Stonyhurst longitude |

| | | |
|---|---|---|
| EF_AxisOrientationUnit | string | Axis oriention unit (deg) |
| EF_AxisLength | float | Axis length of emerging flux pair at completion |
| EF_PosEquivRadius | float | Equivalent radius (i.e. sqrt(area/pi) of positive polarity at completion |
| EF_NegEquivRadius | float | Equivalent radius (i.e. sqrt(area/pi) of negative polarity at completion |
| EF_LengthUnit | string | Units for EF_AxisLength |
| EF_AspectRatio | float | (EF_PosEquivRadius+EF_NegEquivRadius)/(2*EF_AxisLength) |
| EF_ProximityRatio | float | (EF_PosEquivRadius-EF_NegEquivRadius)/(2*EF_AxisLength) |
| MaxMagFieldStrength | float | Maximum magnetic field strength |
| MaxMagFieldStrengthUnit | string | Units for maximum magnetic field strength (e.g. 'G' or 'Mx/cm2') |
| Outflow_Length | float | Length of outflow (can be for CoronalJet or SpraySurge) |
| Outflow_LengthUnit | string | Units for length of outflow (e.g. 'arcsec') |
| Outflow_Width | float | Width of outflow |
| Outflow_WidthUnit | string | Units for width of outflow (e.g. 'arcsec') |
| Outflow_Speed | float | Outflow speed of outflow |
| Outflow_TransSpeed | float | Transverse speed relative to outflow direction |
| Outflow_SpeedUnit | string | Units for speed (e.g. 'km/s') |
| Outflow_OpeningAngle | float | Opening angle of outflow (in degrees) |
| OBS_DataPrepURL | string | URL pointing to information about how data was reduced |
| FL_PeakFlux | float | Flux at peak time |
| FL_PeakFluxUnit | string | Peak flux unit (e.g. erg/cm^2/s) |
| FL_PeakTemp | float | Temperature at peak time |
| FL_PeakTempUnit | string | Unit for FL_PeakTemp (K) |
| FL_PeakEM | float | Emission measure at peak time |
| FL_PeakEMUnit | string | Unit for FL_PeakEM |
| FL_EFoldTime | float | Flare e-folding time |
| FL_EFoldTimeUnit | string | Unit for FL_EFoldTime (s) |
| FL_Fluence | float | Fluence of flare |
| FL_FluenceUnit | string | Unit for FL_Fluence (e.g. erg/cm^2) |
| FL_HalphaClass | string | H Alpha classification (used on some Yohkoh HXT) |
| CD_Area | float | Maximum area of dimming |
| CD_AreaUncert | float | Uncertainty of CD_Area |
| CD_AreaUnit | string | Unit for CD_Area ('millihemisphere' or 'steradians')' |
| CD_Volume | float | Volume of dimming region |
| CD_VolumeUncert | float | Uncertainty of CD_Volume |
| CD_VolumeUnit | string | Units for CD_Volume (e.g. 'cm^3') |
| CD_Mass | float | Mass estimate calculated for dimming event |
| CD_MassUncert | float | Uncertainty for CD_Mass |
| CD_MassUnit | string | Unit for CD_Mass |
| FI_Length | float | Length of Filament spine |

| | | |
|---|---|---|
| FI_LengthUnit | string | Unit for FI_Length |
| FI_Tilt | float | Mean tilt angle (in degrees) of the Filament spine (given as Skel_ChainCode) w.r. to solar equator |
| FI_BarbsTot | integer | Total number of filament barbs |
| FI_BarbsR | integer | Total number of Right Bearing Barbs |
| FI_BarbsL | integer | Total number of Left Bearing Barbs |
| FI_Chirality | integer | Chirality for filament (-1 for sinistral; +1 for dextral; 0 for ambiguous/uncertain) |
| FI_BarbsStartC1 | string | List of Coord 1 of beginnings (closest to spine) of barbs (list delimited by commas for each numbered barb. e.g. 'x1, x2, x3') |
| FI_BarbsStartC2 | string | List of Coord 2 of beginnings (closest to spine) of barbs (list delimited by commas for each numbered barb. e.g. 'y1, y2, y3') |
| FI_BarbsEndC1 | string | List of Coord 1 of ends of barbs (list delimited by commas for each numbered barb. e.g. 'x1, x2, x3') |
| FI_BarbsEndC2 | string | List of Coord 2 of ends of barbs (list delimited by commas for each numbered barb. e.g. 'y1, y2, y3') |
| SG_Shape | string | Shape of sigmoid |
| SG_Chirality | integer | Chirality of sigmoid (-1 for sinistral; +1 for dextral; 0 for ambiguous/uncertain) |
| SG_Orientation | float | Angular rotation (ccw in degrees) of the main axis of the sigmoid to the active region |
| SG_AspectRatio | float | TBD |
| SG_PeakContrast | float | TBD |
| SG_MeanContrast | float | TBD |
| OBS_FirstProcessingDate | string | Earliest date of all images considered part of the event |
| OBS_LastProcessingDate | string | Latest date of all images considered part of the event |
| OBS_LevelNum | float | Level of Data (e.g. 1.5) = LVL_NUM |
| OBS_IncludesNRT | string | T if any image in the event has the NRT flag (bit 30 in QUALITY), "F" otherwise |
| SS_SpinRate | float | Spin/Rotation rate of sunspots |
| SS_SpinRateUnit | string | Unit Spin/Rotation rate of sunspots (e.g. Deg/day) |
| CC_MajorAxis | float | Length of major axis of elliptical fit to cavity |
| CC_MinorAxis | float | Length of minor axis of elliptical fit to cavity |
| CC_AxisUnit | string | Unit of measure for major and minor axes. Typically Rsun |
| CC_TiltAngleMajorFromRadial | float | Angle between major axis and local disk-projected radial vector, measured clockwise from radial vector. |
| CC_TiltAngleUnit | string | Unit of measure for CC_TiltAngleMaorFromRadial. Typicall deg. |
| TO_Shape | string | Apparent shape of topoogical object. Example values can be X-point, cusp, dome, line. |
| UnsignedFlux | float | Total unsigned flux in region |

| | | |
|---|---|---|
| MagFluxUnit | string | Magnetic Flux unit (e.g. 'Mx') |
| MeanInclinationGamma | float | Mean inclination angle (gamma) (degrees) |
| MeanGradientTotal | float | Mean value of the total field gradient |
| MeanGradientVert | float | Mean value of the vertical field gradient |
| MeanGradientHorz | float | Mean value of the horizontal field gradient |
| GradientUnit | string | Gradient unit (e.g. 'G/m') |
| MeanVertCurrentDensity | float | Mean vertical current density |
| CurrentDensityUnit | string | Current density unit (e.g. 'mA/m^2') |
| UnsignedVertCurrent | float | Total unsigned vertical current |
| CurrentUnit | string | Current unit (e.g. 'A') |
| MeanTwistAlpha | float | Mean twist parameter (alpha) |
| TwistUnit | string | Twist unit (e.g. '1/Mm') |
| MeanCurrentHelicity | float | Mean current helicity |
| UnsignedCurrentHelicity | float | Total unsigned current helicity |
| AbsNetCurrentHelicity | float | Absolute value of the net current helicity |
| CurrentHelicityUnit | string | Current helicity unit (e.g. '(G^2)/m') |
| SAVNCPP | float | Sum of the Absolute Value of the Net Currents Per Polarity (SAVNCPP) |
| MeanPhotoEnergyDensity | float | Mean photospheric excess magnetic energy density |
| MeanEnergyDensityUnit | string | Energy Density unit (e.g. 'erg/(cm^3)') |
| TotalPhotoEnergyDensity | float | Total photospheric magnetic energy density (TOTPOT in SHARP) |
| TotalEnergyDensityUnit | string | Energy/length unit (e.g. 'erg/cm') |
| TotalPhotoEnergy | float | Total photospheric magnetic energy (TOTPOT in SHARP times pixel width) |
| TotalPhotoEnergyUnit | string | Energy unit (e.g. 'erg') |
| MeanShearAngle | float | Mean shear angle for B_total (degrees) |
| HighShearAreaPercent | float | Area with shear angle greater than 45 as a percent of total area |
| HighShearArea | float | Area with shear angle greater than 45 as a percent of total area |
| HighShearAreaUnit | string | Area unit (e.g. 'm^2') |
| Log_R_Value | float | Log (Unsigned Flux R) (prelog value in Gauss*MDI-pixels - Schrijver 2007) |
| GWILL | float | Gradient-weighted inversion-line length (Mason & Hoeksema 2010) |
| GWILLUnit | string | GWILL (length) unit (e.g. 'Mm') |
| AR_AxisLength | float | Axis length (bipole separation distance) |
| AR_LengthUnit | string | Units for AR_AxisLength and AR_NeutralLength |
| AR_SumPosSignedFlux | float | Total positive signed flux |
| AR_SumNegSignedFlux | float | Total negative signed flux (negative number) |
| AR_NeutralLength | float | The total length of polarity separation line segments within the AR |

| | | |
|---|---|---|
| AR_PILCurvature | float | Dimensionless curvature of main PIL (PIL length / direct distance between endpoints) |

## 8. Implementation

The database is implemented in PostGRESQL 9.2 with the spatial extender PostGIS 2.1. A Java program is written that can parse JSON files and insert data in the database on the defined schema. That's why the tables *Generic_events*, *AR*, *CH*, *SG* and *FL* are populated with the real data, while the tables *trajectory*, *forms*, *future_event* and *is_followed_by* are populated with artificial data.

In this section, the CREATE TABLE queries and the SELECT queries are written.

### 8.1 CREATE TABLE Statements

Along with the CREATE TABLE statements, statements for creating index, constraint and view are given here.

```
DROP TABLE IF EXISTS is_followed_by;
DROP TABLE IF EXISTS forms;
DROP TABLE IF EXISTS ar;
DROP TABLE IF EXISTS ch;
DROP TABLE IF EXISTS sg;
DROP TABLE IF EXISTS fl;
DROP TABLE IF EXISTS ge;
DROP TABLE IF EXISTS trajectory;
DROP TABLE IF EXISTS FUTURE_EVENT;

CREATE TABLE ge (
 kb_archivid text,
  comment_count NUMERIC,
  gs_thumburl text,
  frm_humanflag boolean,
  event_coordsys text,
  obs_levelnum text,
  event_npixels text,
  gs_imageurl text,
  ar_polarity text,
  frm_paramset text,
  ar_mtwilsoncls text,
  event_starttime TIMESTAMP,
  event_type text,
  intensmin text,
  sol_standard text,
  obs_meanwavel text,
  frm_url text,
```

```
noposition boolean,
active boolean,
intensmax text,
frm_versionnumber float,
area_uncert text,
hpc_geom text,
obs_dataprepurl text,
chaincodetype text,
intensmedian text,
obs_channelid text,
ar_noaaclass text,
event_clippedspatial text,
event_avg_rating text,
eventtype numeric,
intensunit text,
ref_type_0 text,
event_mapurl text,
frm_contact text,
ar_penumbracls text,
intensmean text,
bound_ccstartc1 text,
area_atdiskcenter text,
frm_name text,
frm_identifier text,
obs_observatory text,
event_description text,
boundbox_c2ur float,
obs_firstprocessingdate text,
boundbox_c2ll float,
frm_institute text,
refs_orig text,
ar_mcintoshcls text,
bound_ccstartc2 text,
event_maskurl text,
gs_movieurl text,
event_score text,
event_expires text,
event_probability text,
intensvar text,
frm_daterun TIMESTAMP,
event_coordunit text,
hpc_y float,
hpc_x float,
ref_url_0 text,
ar_numspots text,
kb_archivdate TIMESTAMP,
```

```
kb_archivist text,
intenstotal text,
intensskew text,
obs_includesnrt text,
rasterscan text,
obs_wavelunit text,
ar_noaanum text,
area_atdiskcenteruncert text,
boundbox_c1ur text,
boundbox_c1ll text,
event_importance_num_ratings text,
ar_compactnesscls text,
event_testflag boolean,
event_c2error float,
hrc_r float,
hgs_y float,
obs_title text,
hgs_x float,
hcr_checked boolean,
frm_specificid text,
event_title text,
obs_instrument text,
event_c1error numeric,
revision numeric,
event_endtime TIMESTAMP,
ref_name_0 text,
event_importance text,
event_coord2 float,
event_coord3 float,
event_coord1 float,
area_raw text,
concept text,
event_pixelunit text,
hgc_x float,
hrc_a float,
hgc_y float,
gs_galleryid text,
ar_zurichcls text,
bound_ccnsteps text,
intenskurt text,
event_clippedtemporal text,
rasterscantype text,
area_unit text,
obs_lastprocessingdate text,
PRIMARY KEY (kb_archivid)
);
```

```
SELECT AddGeometryColumn('ge', 'hgc_bbox',4326,'Polygon',2);
SELECT AddGeometryColumn('ge', 'hgc_coord',4326,'Point',2);

SELECT AddGeometryColumn('ge', 'hgs_bbox',4326,'Polygon',2);
SELECT AddGeometryColumn('ge', 'hgs_coord',4326,'Point',2);

SELECT AddGeometryColumn('ge', 'hpc_bbox',4326,'Polygon',2);
SELECT AddGeometryColumn('ge', 'hpc_coord',4326,'Point',2);

SELECT AddGeometryColumn('ge', 'hrc_bbox',4326,'Polygon',2);
SELECT AddGeometryColumn('ge', 'hrc_coord',4326,'Point',2);


SELECT AddGeometryColumn('ge', 'hgc_boundcc',4326,'Polygon',2);
SELECT AddGeometryColumn('ge', 'hgs_boundcc',4326,'Polygon',2);
SELECT AddGeometryColumn('ge', 'hpc_boundcc',4326,'Polygon',2);
SELECT AddGeometryColumn('ge', 'hrc_boundcc',4326,'Polygon',2);

SELECT AddGeometryColumn('ge', 'bound_chaincode',4326,'Polygon',2);

CREATE INDEX event_type_timestamp_index ON ge (event_starttime,
event_type);
CREATE INDEX hpc_coord_index ON ge USING GIST (hpc_coord);

ALTER TABLE ge ADD CONSTRAINT check_time CHECK (event_starttime <
event_endtime);
ALTER TABLE ge ADD CONSTRAINT check_duration CHECK (EXTRACT(day FROM
(event_endtime - event_starttime))<=6);

CREATE TABLE ar(
  kb_archivid text REFERENCES ge (kb_archivid),
  meanphotoenergydensity text,
  ar_spotareepruncert text,
  meaninclinationgamma text,
  currenthelicityunit text,
  totalenergydensityunit text,
  meantwistalpha text,
  absnetcurrenthelicity text,
  meancurrenthelicity text,
  highsheareaunit text,
  currentunit text,
  maxmagfieldstrength text,
  totalphotoenergy text,
  ar_spotarearepr text,
```

```
  totalphotoenergydensity text,
  ar_sumnegsignedflux text,
  gwillunit text,
  ar_neutrallength text,
  ar_pilcurvature text,
  ar_spotareaeprunit text,
  meanshearangle text,
  ref_name_1 text,
  ar_spotarearawuncert text,
  meanenergydensityunit text,
  ref_type_1 text,
  meangradienttotal text,
  ar_sumpossignedflux text,
  meanvertcurrentdensity text,
  meangradientvert text,
  ar_axislength text,
  ar_spotarearaw text,
  unsignedflux text,
  highsheareaapercent text,
  magfluxunit text,
  maxmagfieldstrengthunit text,
  sharp_noaa_ars text,
  log_r_value text,
  totalphotoenergyunit text,
  ar_lengthunit text,
  meangradienthorz text,
  twistunit text,
  gwill text,
  ar_spotarearawunit text,
  gradientunit text,
  currentdensityunit text,
  highsheararea text,
  unsignedcurrenthelicity text,
  savncpp text,
  ref_url_1 text,
  unsignedvertcurrent text,
  CONSTRAINT ar_pkey PRIMARY KEY (kb_archivid)
);
CREATE TABLE ch
(
 kb_archivid text REFERENCES ge (kb_archivid),
  skel_chaincode text,
  skel_nsteps text,
  hgs_skeletoncc text,
  skel_startc2 text,
  hrc_skeletoncc text,
```

```
  skel_startc1 text,
  hgc_skeletoncc text,
  hpc_skeletoncc text,
  CONSTRAINT ch_pkey PRIMARY KEY (kb_archivid)
);
CREATE TABLE sg
(
 kb_archivid text REFERENCES ge (kb_archivid),
  skel_chaincode text,
  skel_nsteps text,
  skel_startc2 text,
  skel_startc1 text,
  sg_meancontrast text,
  sg_aspectratio text,
  sg_chirality text,
  sg_peakcontrast text,
  ref_name_1 text,
  ref_type_1 text,
  event_peaktime text,
  skel_curvature text,
  ref_url_1 text,
  sg_shape text,
  sg_orientation text,
  CONSTRAINT sg_pkey PRIMARY KEY (kb_archivid)
);
CREATE TABLE fl(
 kb_archivid text REFERENCES ge (kb_archivid),
  fl_peakflux text,
  event_peaktime text,
  skel_curvature text,
  fl_fluence text,
  fl_goescls text,
  skel_chaincode text,
  skel_nsteps text,
  skel_startc2 text,
  fl_peakemunit text,
  skel_startc1 text,
  fl_efoldtime text,
  fl_fluenceunit text,
  fl_efoldtimeunit text,
  fl_peakfluxunit text,
  fl_halphaclass text,
  fl_peaktempunit text,
  fl_peakem text,
  ref_name_3 text,
  ref_name_2 text,
```

```
  ref_type_3 text,
  ref_name_1 text,
  ref_type_2 text,
  ref_type_1 text,
  fl_peaktemp text,
  ref_url_2 text,
  ref_url_3 text,
  ref_url_1 text,
  CONSTRAINT fl_pkey PRIMARY KEY (kb_archivid)
);

CREATE TABLE trajectory(
trajectory_id text PRIMARY KEY
);

CREATE TABLE forms(
trajectory_id text REFERENCES trajectory (trajectory_id),
kb_archivid text REFERENCES ge (kb_archivid),
CONSTRAINT forms_pkey PRIMARY KEY (kb_archivid,trajectory_id)
);

Create TABLE future_event(
  fe_id text PRIMARY KEY,
  event_type text not null,
  probability numeric,
  time_of_gen TIMESTAMP,
  event_coord1 float,
  event_coord2 float
);

Create table is_followed_by(
  fe_id text REFERENCES future_event (fe_id),
  trajectory_id text REFERENCES trajectory (trajectory_id),
  CONSTRAINT is_followed_by_pkey PRIMARY KEY (fe_id,trajectory_id)
);
```

### 8.2 SELECT Queries

Here, some example SELECT queries are shown.

1. Show all trajectory ids form the *trajectory* table.

```
dbproject=# SELECT * FROM trajectory;
 trajectory_id
---------------
 traj_1
 traj_2
 traj_3
 traj_4
(4 rows)
```

2. Show all entries in the *forms* table.

```
dbproject=# SELECT * FROM forms;
 trajectory_id |                        kb_archivid
---------------+------------------------------------------------------------
 traj_1        | SG_SigmoidSniffer_20120103_195307_2012-01-01T02:52:57.620_0
 traj_1        | SG_SigmoidSniffer_20120103_195549_2012-01-01T03:52:57.620_0
 traj_1        | SG_SigmoidSniffer_20120103_200222_2012-01-01T06:22:57.620_1
 traj_2        | CH_SPoCA_20130601_055223_20130601T053555_0
 traj_2        | CH_SPoCA_20130601_055223_20130601T053555_3
 traj_3        | AR_NOAASWPCObserver_20120412_233404_NOAA11413_20120201
 traj_3        | AR_NOAASWPCObserver_20120412_233404_NOAA11411_20120201
 traj_3        | AR_NOAASWPCObserver_20120412_233404_NOAA11410_20120201
 traj_3        | AR_SPoCA_20120201_070500_20120201T064913_3
 traj_3        | AR_SPoCA_20120201_070500_20120201T064913_9
(10 rows)
```

3. Show all future event id, event type, probability and event coordinates in *future_event* table.

```
dbproject=# SELECT fe_id, event_type, probability, event_coord1, event_coord2 FR
OM future_event;
 fe_id  | event_type | probability | event_coord1 | event_coord2
--------+------------+-------------+--------------+--------------
 f_SG_1 | SG         |         0.8 |    -15.12704 |     -26.0704
 f_SG_2 | SG         |         0.5 |    -20.12704 |     -36.0704
 f_CH_1 | CH         |        0.75 |    -65.12704 |     -28.0704
(3 rows)
```

4. Show all entries in the table *is_followed_by*.

```
dbproject=# SELECT * FROM is_followed_by;
 fe_id  | trajectory_id
--------+---------------
 f_SG_1 | traj_1
 f_SG_2 | traj_1
 f_CH_1 | traj_2
(3 rows)
```

5. Show all event coordinates of all the trajectories.

```
dbproject=# SELECT forms.trajectory_id, ge.event_coord1, ge.event_coord2 FROM ge
 JOIN forms ON ge.kb_archivid = forms.kb_archivid;
 trajectory_id | event_coord1 | event_coord2
---------------+--------------+--------------
 traj_1        |     -9.12704 |     -18.4901
 traj_1        |     -8.54403 |     -18.4837
 traj_1        |      28.7842 |     -20.7368
 traj_2        |     -773.395 |     -193.825
 traj_2        |      164.974 |      357.188
 traj_3        |           -8 |            8
 traj_3        |           28 |          -27
 traj_3        |          -11 |           18
 traj_3        |      736.507 |     -188.263
 traj_3        |      161.053 |      444.881
(10 rows)
```

6. Show the event coordinates and sg_chirality of the SG event instance whose kb_archivid is "SG_SigmoidSniffer_20120103_200222_2012-01-01T06:22:57.620_1".

```
dbproject=# SELECT ge.event_coord1, ge.event_coord2, sg.sg_chirality FROM ge,sg
WHERE ge.kb_archivid = sg.kb_archivid AND sg.kb_archivid LIKE 'SG_SigmoidSniffer
_20120103_200222_2012-01-01T06:22:57.620_1';
 event_coord1 | event_coord2 | sg_chirality
--------------+--------------+--------------
      28.7842 |     -20.7368 | 0
(1 row)
```

7. Count the instances of each event type in May, 2012.

```
SELECT event_type, count(*) as event_count_by_type from ge WHERE
event_starttime > '2012-05-01 01:14:47' AND event_endtime < '2012-06-01
01:14:47' GROUP BY event_type;

 event_type | event_count_by_type
------------+---------------------
 SG         |                 489
 FL         |                2332
(2 rows)

Time: 35.723 ms
```

8. Spatial query which shows performance improvement while using indexes.
Query: Count all intersecting event instances.
i) Following query is performed on non-indexed attribute.

```
select count(*) from ge as g1, ge as g2 where ST_intersects(g1.hrc_bbox,
g2.hrc_bbox) AND g1.event_starttime < '2012-01-10 04:45:00'::timestamp;
 count
--------
 471984
(1 row)

Time: 25494.026 ms
```

ii) Following query is performed on indexed attribute.

```
select count(*) from ge as g1, ge as g2 where ST_intersects(g1.hpc_bbox,
g2.hpc_bbox) AND g1.event_starttime < '2012-01-10 04:45:00'::timestamp;
 count
--------
 470178
(1 row)

Time: 9618.431 ms
```

9. Show the number of records used in the testing.

```
select count(*) from ge;
 count
-------
 18135
(1 row)

Time: 6.081 ms
```

## 9. Conclusion

In this project, almost all concepts learnt by us in the class CSC 6710 are implemented. Our design is comparatively storage efficient that HEK. While HEK uses (169 + 128 + 147 + 135) = 579 attributes for storing four event types AR, CH, FL and SG, our design uses (120 + 49 + 8 +27 + 15) = 219 attributes. Better performance can be achieved, if we remove "unnecessary" attributes, which holds only null values most of the time. In future, we plan to implement graphical user interface for searching of solar events. We also plan to include all 22 event types in our database.

## 10. Acknowledgement

We express our deepest gratitude to Berkay Aydin, Senior PhD student, Department of Computer Science, Georgia State University for his constant guidance throughout the project. Also, we thank the GSU Honors College and Computer Science Department for additional support.

## 11. Individual Roles

| Contribution | Person |
|---|---|
| Weekly meeting with Berkay | All |
| Requirement analysis for spatial queries | Hamdi |
| Requirement analysis for temporal queries | Ahmet |
| Requirement analysis for spatiotemporal queries | Chase |
| Requirement analysis for non-spatiotemporal queries | Soukaina |
| Java program for attribute analysis | Ahmet |
| R program for attribute analysis | Hamdi |
| Java program for finding common attributes | Soukaina |
| EER Model | Hamdi |
| Relational Model | Hamdi |
| Creating github repository for shared development | Ahmet |
| Writing CREATE TABLE statements | Soukaina |
| Java program for fetching data from JSON to database | Ahmet |
| Indexing, creating constraints and views | Ahmet |
| Query generating and testing | All |
| Making presentation | Hamdi |
| Writing final project report | Chase, Hamdi |

**Signatures of each member:**

Ahmet Kucuk

Andre Kenneth Chase Randall

Shah Muhammad Hamdi