# The Sequoia Schema: Example and Strategy

Jim Frew
Mike Olson

Sequoia 2000 Review
03 March 1993

# A Canonical Query

- have satellite images stored as large objects
- want a specific geographic subset
- want to specify the subset in lat-lon, regardless of the image coordinate system
- want the corresponding subsets saved as new large objects, for access by existing programs

# Example Implementation

- load AVHRR satellite images as inversion large objects (i.e. accessible as files)
- create ARRAY type so Postgres can access the innards of the large objects
- create GEOLOC type to store geographic coordinates
- write functions to support latitude-longitude and Lambert Azimuthal Equal Area coordinate systems
- create AVHRR class to bind image ARRAYs to corner GEOLOCs

# Arrays as Large Objects

- generic: multidimensional; multiple internal formats

```
typedef struct {
    char *format;        // internal format name
    int   element_size; // # bytes per element
    int   n_dimensions; // # dimensions
    int  *dimension;    // [1..n_dimensions]
    char *name;          // large object name
} ARRAY;

class ARRAY_FORMAT (
    format   = char16,  // internal format name
    clipfunc = regproc, // clip function
    ...                 // (other functions)
)
```

# Geographic Locations as Polymorphic Types

- GEOLOC supports multiple coordinate systems; comparisons done in "standard" coordinates

```
typedef struct {
char  *name;          // coordinate system name
    float  precision;   // precision of (x,y)
    double x;           // x (usu. E) coordinate
    double y;           // y (usu. N) coordinate
    // state            // parameters (e.g. zone)
} GEOLOC;

create GEO_UNITS (
    name     = char8,   // coordinate system name
    from_std = regproc, // convert to std coords
    to_std   = regproc  // convert from std coords
)
```

# AVHRR Images as a Class

- class with corner coords as GEOLOCs and the image itself as an ARRAY

```
create AVHRR (
    image_nw   = GEOLOC,
    image_se   = GEOLOC,
    image      = ARRAY
)
```

- clip AVHRR images to a specified geographic rectangle

```
define function clip_avhrr
    (language = "c", returntype = ARRAY)
    arg is (AVHRR, format, GEOLOC, GEOLOC)
    as "avhrr.o"
```

# The $2^16 Query

- retrieve a 2 by 4 degree subscene from all relevant AVHRR images

```
retrieve (images =
    clip_avhrr(
        AVHRR,                      // class to search
        "pgm",                      // output PGM format
        "latlon, 1, -122, 38", // NW: Mt. Diablo
        "latlon, 1, -120, 34", // SE: Santa Barbara
    )
)
```

- output includes large object name(s), so you can:

```
pcat {name} | xv -  # to display in an X window
```

# Strategy for Schema Development

Repeat this exercise for an exemplary dataset of each fundamental data type used by S2K Earth scientists:

- image (e.g. AVHRR)
- polygon (e.g. global political boundaries)
- tabular (e.g. weather station measurements)
- multidimensional field (e.g. GCM output)
- text (e.g. technical reports)

Goal: support single queries across multiple data types