

Antibody Database Schema

```
create temporary module ig_mod

declare structure --> entity
declare protein_code(structure) -> string      % PDB
declare protein_name(structure) -> string      % PDB
declare source(structure) -> string            % PDB
declare authors(structure) -> string          % PDB
declare resolution(structure) -> float        % PDB
declare ig_name(structure) -> string          % By hand ("NEWM")
declare class(structure) -> string            % By hand ("IgG")
key_of structure is protein_code

declare chain --> entity
declare component_protein(chain) -> structure % PDB
declare component_id(chain) -> string         % PDB
declare pdb_chain_id(chain) -> string        % PDB
declare num_residues(chain) -> integer       % PDB
declare chain_class(chain) -> string         % By hand
key_of chain is key_of(component_protein), component_id

declare residue --> entity
declare pos(residue) -> integer              % PDB
declare name(residue) -> string              % PDB
declare has_component(residue) -> chain      % PDB
declare residue_chain(residue) -> chain      % Test: GJLK 11/7/00
declare author_id(residue) -> string         % PDB
declare kabat_position(residue) -> string    % Kabat file (####.DAT)
declare omega(residue) -> float             % Test: GJLK 11/7/00
key_of residue is key_of(has_component), pos

declare atom --> value_entity
declare x(atom) -> float                     % PDB
declare y(atom) -> float                     % PDB
declare z(atom) -> float                     % PDB
declare accessibility(atom) -> float         % Default 0.0 used.
declare atom(residue, string) -> atom       % PDB

declare ig_domain --> entity
declare domain_structure(ig_domain) -> structure
declare name(ig_domain) -> string            % By hand (VL,VH,CL,CH1-3)
declare ig_domain_chain(ig_domain) -> chain
declare domain_type(ig_domain) -> string     % By hand ("variable", ...)
declare source(ig_domain) -> string          % By Hand ("human", "mouse")
declare subgroup(ig_domain) -> string        % By Hand ("lambda I")
declare start(ig_domain) -> integer
declare end(ig_domain) -> integer
%declare kabat_entry(ig_domain) -> kabat_entry % By hand ("HLL10J")
key_of ig_domain is key_of(domain_structure), name

declare structural_cdr --> entity
declare structural_cdr_domain(structural_cdr) -> ig_domain
declare name(structural_cdr) -> string
declare start(structural_cdr) -> integer
declare end(structural_cdr) -> integer
declare canonical_conformation(structural_cdr) -> integer
key_of structural_cdr is key_of(structural_cdr_domain), name

declare structural_framework --> entity
declare structural_framework_domain(structural_framework) -> ig_domain
declare name(structural_framework) -> string
declare start(structural_framework) -> integer
declare end(structural_framework) -> integer
key_of structural_framework is key_of(structural_framework_domain), name

declare kabat_cdr --> entity
declare kabat_cdr_domain(kabat_cdr) -> ig_domain % Method
declare name(kabat_cdr) -> string                % Method
declare start(kabat_cdr) -> integer              % Method
declare end(kabat_cdr) -> integer                % Method
key_of kabat_cdr is key_of(kabat_cdr_domain), name

declare kabat_framework --> entity
declare kabat_framework_domain(kabat_framework) -> ig_domain
declare name(kabat_framework) -> string
declare start(kabat_framework) -> integer
declare end(kabat_framework) -> integer
key_of kabat_framework is key_of(kabat_framework_domain), name

declare absolute_pos(chain, integer) -> residue % PDB
declare res_by_name(chain, string) -> residue
declare kabat_residue(ig_domain, string) -> residue;
```