MoAB Database:

- Mesh (connectivity & adjacency)
- Tags on mesh

RefEntity (dense tag storage):
- **BitRefinmentLevel**

Field meshset:
- **name of the field**
- **ID of space (H1,Hdiv,etc.)**
- **ID of approximation base (Legendre,etc.)**
- **Coeoffincnts number (**rank**)**
- **Coordinate systems (**reference and current base**)**

FieldEntity (sparse or dense storage):
- **field order of approx.**
- **filed DOFs values**

ptr to moab tags

Field multi-index

Shared pointer to container of field structures (not many of those).
In structure:
- ID & pointers to internal MOAB tags storage
- sequences (vectors) of field entities/dofs structures

interface<PTR>

FieldEntity seq. 0 | **FieldEntity** seq. 1

.. | **FieldEntity** Seq. N

DOFs Seq. 0 | ..

DOFs Seq. 2

inheritance by pointer in interface

FieldEntity multi-index

Aliased shared pointer to element of sequence container of FieldEntities.
- ID (owner proc | EntityHandle | Field ID)
- sequence to dofs on entity (wihch are not in Field data structure)
- approx. order & tag ptr. to field data on mesh

interface<PTR>

DOFs sequence

inheritance by pointer in interface

Dof multi-index

Aliased shared pointer to container of dofs structures (large number of those).
In structure:
- ID (dof number on entity | UId of FieldEntity)
- Shared pointer to **FieldEntity Interface**

interface<PTR>

Aliased share pointer to sequence

```
FieldEntity by aliased shared pointer:
shared_ptr<vector<FieldEntity> > seq0;
entity_n = shared_ptr<FieldEntity>(&seq0[n],seq0);

FieldEntity by aliased shared pointer:
shared_ptr<vector<Dof> > seq0;
dof_n = shared_ptr<Dof>(&seq0[n],seq0);
```

* Vector of FieldEntities/Dofs is destroyed when all elements in sequence are destroyed. Memory is allocated in sequences (blocks) to minimalist set-up/build database time.