08 – Database Storage Models & Layout

1. Type Representation
    A. Fixed length fields : stored in byte array itself
    B. Variable length fields : since pointer size is fixed, store pointer in byte array, real contents are stored in separate data blocks.
    C. Null data types
        i. Using special values to represent NULL
        ii. Column Header to indicate that attributes are NULL
        iii. Per attribute Flag to indicate that attributes are NULL
        (stupid because of attribute padding)

2. Data Layout/Alignment
    A. Word alignment for get data faster, without additional read or operation
    B. Approach
        i. Use padding to align attributes with word
        ii. Reorder the attributes to reduce padding

3. Storage Models

    A. N-ary Storage Model

        i. Row-oriented storage model
        all attributes of single tuple are stored contiguously

        ii. Ideal to OLTP workloads (operates on single entity at a time)

        iii. Can use index oriented physical storage

        iv. Not good for OLAP workloads (operates on many entity)

    B. Decomposition Storage Model

        i. Column-oriented storage model
        single attributes of all tuples are stored contiguously

        ii. Ideal to OLAP workloads (operates on many entity but less attribute)

        iii. Not good for point query (operates on single entity, many attributes)

        iv. Design policy

            1. Tuple identification

                A. Each attributes has fixed length

                B. Each value is stored with tuple id

            2. Data organization

                A. Insertion order (just at free slot)

                B. Sorted order (order by some rules for tuple)

                C. Partitioned (partition by some rules for attribute value)

    C. Hybrid Storage Model

        i. Manage different tuples by access pattern

        ii. Flexible single machine vs. separated machine

4. System Catalogs
    A. Store information about tables
    B. Specialized code for bootstrapping catalog tables.
       because of chicken-egg problem
    C. Catalog information
        i. Schema changes
            1. Add column
                A. NSM : copy all tuples and add new attributes
                B. DSM : just add new segment
            2. Drop column
                A. NSM : copy all tuples and drop attributes
                   or just set bit for deprecate attributes
                B. DSM : just drop segment
            3. Change column
        ii. Index information
        iii. Sequences (global counter for “auto-increment”…, etc)