15 – Query Planning & Optimization 2

1. Plan Cost Estimation
    A. Selection cardinality : avg number of records for an some attribute

    B. Selectivity : fraction of records that satisfy predicate.
        i. Use selectivity to estimate # of records that satisfy predicate (negation, disjunction, conjunction, range search….)
        ii. There are three assumption
            1. Uniform data – if data is non-uniform, use quantiles
            2. Independent predicate – correlated predicate can be a problem
            3. Inclusion principle
        iii. Sampling
            1. Collect samples to estimate selectivity
            2. Sample update occur when table changes significantly

2. Plan Enumeration

    A. OLTP Query Planning

        i. Easy, because it is search argument able

    B. Multi-relation Query Planning

        i. Only consider left deep trees – fully pipelined plan

    C. Process (do dynamic programming)

        i. Orderings

        ii. Plans for each operator

        iii. Access method for each table

    D. Postgres optimizer

        i. Traditional dynamic programming approach : explained above

        ii. Genetic query optimizer

            1. Generate random plan, throw away worst plan

            2. Mix plans remaining, then throw worst plan again

            3. Repeat *2*, until newly generated plans are not new ones

            4. Pick plan which cost is minimum that appeared at process above

3. Nested Sub-queries

    A. Re-write to flatten query

    B. Decompose sub-query and store result at temporary table