Homework 3.1

Show the implementation of relational algebra in a Map-Reduce environment.

Selection by Map-Reduce:

The Map Task: For each tuple t in the dataset, test if it satisfies the condition. If so, produce the key-value pair (t, t). That is, both the key and value are t.

The Reduce Task: identify each key-value pair to output the result.

Projection by Map-Reduce:

The Map Task: For each tuple t in the dataset D1, construct a tuple t' by eliminating from t those components whose attributes are not in the dataset D2, and produce the key-value pair (t', t').

The Reduce Task: For each key t' produced by any of the Map tasks, there will be one or more key-value pairs (t', t'), which means the same tuple could appear several times, we must eliminate duplicates. The Reduce task turns (t', [t', t'…t']) into (t', t'), so it produces exactly one pair (t', t') for this key t'.

Union by Map-Reduce:

The Map Task: For each tuple t in the dataset D1 and D2 which have the same schema, produce the key-value pair (t, t).

The Reduce Task: Associated with each key t there will be either one or two values. Produce output (t, t) in either case.

Intersection by Map-Reduce:

The Map Task: For each tuple t in the dataset D1 and D2 which have the same schema, produce the key-value pair (t, t).

The Reduce Task: If the key t has a list of two values [t, t] associated with it, then produce (t, t). However, if the value-list associated with key t is just [t], then one of D1 and D2 is missing t, produce nothing.

Difference by Map-Reduce:

The Map Task: For each tuple t in the dataset D1, produce a key-value pair (t, D1), and for each tuple t in the dataset D2, produce a key-value pair (t, D2).

The Reduce Task: For each key t, if the associated value list is [D1], then produce (t, t). Otherwise, produce nothing.

Join by Map-Reduce:

The Map Task: For each tuple (a, b) in the dataset D1, produce a key-value pair (a, (D1, b)). For each tuple (a, c) in the dataset D2, produce a key-value pair (a, (D2, c)).

The Reduce Task: Each key value a will be associated with a list of pairs that are either of the form (D1, b) or (D2, c). Construct all pairs consisting of one with first component D1 and the other with first component D2, say (D1, b) and (D2, c). The output from this key and value list is a sequence of key-value pairs. The key is irrelevant. Each value is one of the triples (a, b, c) such that (D1, b) and (D2, c) are on the input list of values.

Grouping and Aggregation by Map-Reduce:

Map will perform the grouping, while Reduce does the aggregation.

The Map Task: For each tuple (a, b, c), produce the key-value pair (a, b).

The Reduce Task: Each key a represents a group. Apply the aggregation operator θ to the list [b1, b2…bn] of B-values associated with key a. The output is the pair (a, x), where x is the result of applying θ to the list.