DBMS Digital Assignment - 3

Name: Karan Singh

Reg. No.: 19BCE2310

Slot: D2

Ques. 1 - List the operations of relational algebra and the purpose of each.

Ans.:

Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output. It uses operators to perform queries. An operator can be either unary or binary. They accept relations as their input and yield relations as their output.
The operations of Relational Algebra are as follows:

- ➔ SELECT
- ➔ UNION
- ➔ CARTESIAN PRODUCT
- ➔ PROJECT
- ➔ INTERSECTION
- ➔ DIFFERENCE
- ➔ NATURAL JOIN
- ➔ EQUI JOIN
- ➔ THETA JOIN
- ➔ DIVISION

1. SELECT Operation -
    - This operation is used to obtain a subset of rows or tuples of a relation. It retrieves the tuples satisfying the condition given in the input.
    - It is denoted by the symbol “σ” where the subscript is the condition given for the SELECT operation.
    - Example: $\sigma_{id=1234}$(Teacher) retrieves the tuples from the table Teacher whose id = 1234

2. UNION Operation -
    - This is a binary operation i.e.; it requires two operands.
    - When UNION Operation is applied on two relations A and B, the resultant relation consists of all the tuples in relation A or B or both A and B.
    - If a tuple is common in both A and B then A U B will contain only one of those 2 instances.
    - The symbol used to represent this operation is “U”

3. CARTESIAN PRODUCT Operation -
    - This is a binary operation i.e.; it requires two operands.
    - When CARTESIAN PRODUCT operation is applied on two relations A and B, the resultant relation consists of all the attributes of relation A and B along with all possible combinations of the tuples of those two relations.
    - The Cartesian Product is represented as A □ B.

4. PROJECT Operation -
    - This operation is used to retrieve certain attributes/columns from a given table.
    - The symbol for denoting this operation is “π”.
    - Example: $\pi_{id, name}$(Teacher) - this operation retrieves the columns id and name from the relation Teacher.

5. INTERSECTION Operation -
    - This is also one of the binary operations as it requires two operands.
    - When INTERSECTION operation is applied on relations A and B, the resultant relation consists of only the tuples that are in both A and B.
    - Intersection of two relations is denoted as A ∩ B.

6. DIFFERENCE Operation -
    - This is another one of the binary operations.
    - When DIFFERENCE operation is applied on relations A and B, the resultant relation consists of only the tuples that are A but not in B.
    - The Difference operation between two relations is denoted as A - B.

7. NATURAL JOIN Operation -
    - It combines all the tuples of relations A and B that satisfy the condition but join attributes of relation B are not included in the resultant relation.

    - The notations of NATURAL JOIN between the relations A and B is given as - $A*_{\text{(join condition)}}$ B

8. EQUI JOIN Operation -
    - An EQUIJOIN operation combines all the tuples of relations A and B that satisfy the condition.
    - The notations of EQUI JOIN between the relations A and B is given as -

      $A \bowtie_{\text{(join condition)}} B$

9. THETA JOIN Operation -
    - THETA JOIN operation combines related tuples from two relations and outputs as a single tuple.
    - The symbol used to denote this join is $\bowtie_{\theta}$

10. DIVISION Operation -
    - This combines all the tuples of *A(Z)* that appears in A with every tuple from B(Y) to form a new relation where C = A ∪ B
    - The symbol used to denote DIVISION operation is “÷”.

Ques. 2 - How does the granularity of data items affect the performance of concurrency control? What factors affect selection of granularity size for data items?

Ans.:

In Database systems we refer to Granularity as the size of the data items. Smaller the size of the data item it is “fine granularity”, larger size is “coarse granularity”.

Granularity of data items affect the performance of concurrency control in a few ways.

- The larger the data item size is, the lower will be the degree of concurrency permitted. For example, if the data item size is a disk block, a transaction T1 that needs to lock a record X must lock the whole disk block D that contains X because a lock is associated with the whole data item (block).
  Now, if another transaction T2 wants to lock a different record Y that happens to reside in the same block D in a conflicting lock mode, it is forced to wait. If the

data item size was a single record, transaction S would be able proceed, because it would be locking a different data item.

- The smaller the data item size is, the more the number of items in the database. Because every item is associated with a lock, the system will have a larger number of active locks to be handled by the lock manager.
  More lock and unlock operations will be performed, causing a higher overhead. In addition, more storage space will be required for the lock table. For timestamps, storage is required for the read_TS and write_TS for each item, and there will be similar overhead for handling a large number of items.

- ➔ The **factors affecting selection of granularity size** for data items are discussed below :-
  The number of transactions involved is the main determining factor for the data item size. If a typical transaction accesses a small number of records, it is advantageous to have the data item granularity be one record.
  On the other hand if a transaction typically accesses many records in the same file. It may be better to have a block or file granularity so that the transaction will consider all the records as one data item.