Date ....

Chapter 8
The Relational Algebra and Relational Calculus.

* Relational Algebra

→ The basic set of operations for the relational model is known as the relational algebra. These operations enable a user to specify basic retreival requests.

→ The result is a new relation formed from one or more relations.

→ A sequance of relational algebra operations forms a relational algebra expressions, whose result will also be a relation that represents the result of a database query.

query processing in RDBMS

* Relational Calculus

→ Firm foundation of mathematical logic.

→ Used in high-level declarative language for writing queries.

→ No order of operations in RC expressions

→ SQL is based on RC but also has some feature of RA.

* Unary Relational Operations.

→ SELECT (σ) Operation

- Used for selecting a subset of the tuples from a relation that satisfy a selection condition
- Denoted by $\sigma_{<\text{selection condition}>}(R)$
- σ (sigma) denotes select operator.
- Selection condition is a boolean expression

specified on the attributes of relation R.
example:
Those whose salary is greater than $30,000

$\sigma_{salary>30,000}$ (EMPLOYEE)

- The SELECT operation $\sigma_{<condition>}(R)$ produces a relation S that has same schema as R.
- SELECT operation is commutative.

$$\sigma_{<condition1>}(\sigma_{<condition2>}(R)) = \sigma_{<condition2>}(\sigma_{<condition1>}(R))$$

- A cascaded SELECT operation may be applied in any order.
- A cascaded SELECT operation may be replaced by a single selection with a conjunction of all the conditions.

→ Project (π) Operation

- Selects certain columns from the table and discards the other columns.
- Creates a vertical partioning.
- $\pi_{<attribute\ list>}(R)$
- π (called pi) is used to represent projection
- Projection operation removes any duplicate tuples, so the result of the project operation is a set of tuples and hence a valid relation.
- Example:
  Employees name, salary.
  $\pi_{name,salary}$ (EMPLOYEE)
- Number of tuples in the result of projection is less or equal to the no. of tuples in R.

- If the list of attributes includes a key of R, then the no. of tuples is equal to the no. of tuples in R.
- Projection is not commutative

* Sequence of Operations and the RENAME Operation

→ We can write the operations as a single relational algebra expression by nesting the operations.

$\pi_{Fname, salary}(\sigma_{DNO=5}(EMPLOYEE))$

OR

We can apply one operation at a time and create intermediate result relations, giving a name to it.

$DEP5\_EMPS \leftarrow \sigma_{DNO=5}(EMPLOYEE)$

$Result \leftarrow \pi_{Fname, salary}(DEP5\_EMPS)$.

→ Rename Operation (ρ)

- Operator is ρ (called Rho)
- $\rho_{S(B_1, B_2 -- B_n)}(R)$ is a renamed relation S based on R with column names $B_1, B_2, -- B_n$
- Example:

$\rho_{T<fn, ln, DNO>}(\pi_{<fname, lname, Dnumber>}(EMPLOYEE))$

* Relational Algebra Operations from Set Theory

→ UNION Operation

- The result of this operation, denoted by $R \cup S$, is a relation that includes all tuples that are either in R or in S or in both R and S. Duplicate tuples are eliminated.

- The union produces the tuples that are in R_1 or R_2 or both. The 2 operands must be type compatible (same domain and datatype} and degree)

→ Intersection Operation.

- The result of this operation, denoted by $R \cap S$ is a relation having tuples that are common to the both operand relations. Must be type compatibl

→ Difference.

- The result of this operation, denoted by $R-S$, is a relation that includes all tuples that are in R but not in S. Two operands must be type compatible.

* Both union and intersection are commutative.
  $R \cup S = S \cup R$ and $R \cap S = S \cap R$

* Both union and intersection are treated as n-ary operations and are associative.
  $R \cup (S \cup T) = (R \cup S) \cup T$ and $(R \cap S) \cap T = R \cap (S \cap T)$

* The minus, is not commutative.
  $R-S \neq S-R$.

→ Cartesian (or cross product) operation.

- The cartesian product of 2 relations yields a relation with all possible combinations of the tuples of the 2 relations ~~set~~ operated upon.
- Dnoted by a cross (X)

- The degree of the resultant relation is the sum of the degrees of the 2 relations operated upon.
- The number of tuples (cardinality) of the new relation is the product of the numbers of tuples of the 2 relations.

  $R \times S$, $|R| = n_R$ and $|S| = n_S$

  $\therefore |R \times S| = n_R \times n_S$
- The 2 relations need not be type compatible.

* Binary Relational Operations.

→ JOIN ($\bowtie$) Operation.

- JOIN is denoted as $\bowtie$, used to combine related tuples from 2 relations into single longer tuples.
- It is important for relational database with more than a single relation, bcoz it allows us to process relationships among relations.
- General form of JOIN operation:

  $R(A_1, A_2, \ldots, A_n)$ and $S(B_1, B_2, \ldots, B_m)$ is

  $$R \bowtie_{<join\ condition>} S$$
- Result of JOIN is a relation Q with $n+m$ attributes.

* Difference b/w Cartesian Product & JOIN.

In JOIN, only combinations of tuples satisfying the join condition appear in the result. whereas in Cartesian Product all combinations of tuples are included.

→ Theta Join.

- Each <condition> is of the form $A_i \theta B_j$

  A_i → Attribute of R. } have same domain.

  B_j → Attribute of S.

  θ → One of the comparison operator $\{=,<,\leq,>,\geq,\neq\}$
- JOIN with such general condition is called a Theta JOIN.
- Tuples with NULL join attributes do not appear in the Result.

→ Equijoin Operation. $\bowtie_{a_1=b_1}$

- JOIN operation with equality comparison only.
- In the result we have one or more pair of attributes that have identical values in every tuple.

→ Natural JOIN. (*)

- Denoted as *
- Natural join requires that the 2 joins attributes have the same name

Join Selectivity

- Expected size of join result divided by the maximum size $n_R * n_S$.

→ Inner Joins

- Defined formally as a combination of cartesian product and selection

Date ....

* A complete set of Relational Algebra Operations

- The set operations including σ, π, ∪, ~~∩~~, − and × is called a complete set bcoz any other relational algebra expression can be expressed by a combination of these 5 operations.

* DIVISION Operation

- Denoted by ÷
- DIVISION operation is applied to 2 relations $R(Z) \div S(X)$ where the attributes of S are a subset of the attributes of R ; that is $X \subseteq Z$.
- The result of DIVISION is a relation $T(Y)$ that includes a tuple t if tuples t_R appear in R with $t_R[Y] = t$, and with $t_R[X] = t_S$ for every tuple t_S in S. This means that for ~~combination with every tuple in S.~~ a tuple t to appear in the result T of the DIVISION, the values in t must appear in R in combination with every tuple in S

* Notation for Query Trees.

- A query tree is a tree data structure that corresponds to a relational algebra expression.
- It represents the input relations of query as leaf nodes of the tree.
- It represents the relational algebra operations as internal nodes.
- Also called query expression / evaluation tree.

* Additional Relational Operations.

Generalized Projection

Allows functions ~~of~~ attributes to be included ~~as~~ in

Spiral

the projection list.

$$\pi_{F1, F2, \ldots, Fn}(R)$$

* Aggregate functions and grouping

- Common functions applied to collections of numeric values.
- Includes Sum, Average, Maximum, and minimum. The count function is used for counting tuples an values.
- Use of the functional operator (script F) '$\mathcal{F}$'

$$<\text{grouping attribute}> \mathcal{F} <\text{function list}> (\text{Table})$$

* Left Outer Join

- Denoted by ⟕
- The LEFT OUTER JOIN operation keeps every tuple in the first, or left, relation R in R ⟕ S; if no matching tuple is found in S, then the attributes of S in the join result are filled or padded with NULL values.

* Right Outer Join

- Denoted by ⟖
- It keeps every tuple in the second, or right, relation S in the result of R ⟖ S.

* Full Outer Join

- Denoted by ⟗
- It keeps all tuples in both the left and right relations with no matching tuples are found, padding them with NULL values as needed.

* The Outer Union Operation

- It takes the union of tuples from 2 relations that have some common attributes, but are not union (type) compatible.