The Relational Algebra

Part 1

Unary Relational Operations: SELECT and PROJECT

The SELECT Operation

- The SELECT operation is used to choose a subset of the tuples from a relation that satisfies a **selection condition**.
- In general, the SELECT operation is denoted by
  $\sigma_{\text{<selection condition>}}(R)$
  where the symbol σ (sigma) is used to denote the SELECT operator and the selection condition is a Boolean expression (condition) specified on the attributes of relation R.
- Notice that R is generally a *relational algebra expression* whose result is a relation—the simplest such expression is just the name of a database relation.
- The relation resulting from the SELECT operation has the same attributes as R.

- The Boolean expression specified in `<selection condition>` is made up of a number of clauses of the form

  `<attribute name> <comparison op> <constant value>`

  or

  `<attribute name> <comparison op> <attribute name>`

- $\sigma_{\text{Dno=4}}$ (EMPLOYEE)
- $\sigma_{\text{Salary>30000}}$ (EMPLOYEE)
- $\sigma_{\text{(Dno=4 AND Salary>25000) OR (Dno=5 AND Salary>30000)}}$ (EMPLOYEE)

- The SELECT operator is **unary**; that is, it is applied to a single relation.
- Moreover, the selection operation is applied to *each tuple individually*; hence, selection conditions cannot involve more than one tuple.
- The **degree** of the relation resulting from a SELECT operation—its number of attributes—is the same as the degree of R.
- The number of tuples in the resulting relation is always *less than or equal to* the number of tuples in R.
- That is, $|\sigma_C(R)| \leq |R|$ for any condition C.
- The fraction of tuples selected by a selection condition is referred to as the **selectivity** of the condition.

- Notice that the SELECT operation is **commutative**; that is,

$$\sigma_{\texttt{<cond1>}}(\sigma_{\texttt{<cond2>}}(\texttt{R})) = \sigma_{\texttt{<cond2>}}(\sigma_{\texttt{<cond1>}}(\texttt{R}))$$

- Hence, a sequence of SELECTs can be applied in any order.
- In addition, we can always combine a **cascade** (or **sequence**) of SELECT operations into a single SELECT operation with a conjunctive (AND) condition; that is,

$$\sigma_{\texttt{<cond1>}}(\sigma_{\texttt{<cond2>}}(\ldots(\sigma_{\texttt{<cond}n\texttt{>}}(R))\ \ldots)) =$$
$$\sigma_{\texttt{<cond1> AND<cond2> AND...AND <cond}n\texttt{>}}(R)$$

- In SQL, the `SELECT` condition is typically specified in the `WHERE` clause of a query.

- $\sigma_{\text{Dno=4 AND Salary>25000}}$ (EMPLOYEE)
-
```
SELECT *
FROM   EMPLOYEE
WHERE  Dno=4 AND Salary>25000;
```

The PROJECT Operation

- The **PROJECT** operation selects certain *columns* from the table and discards the other columns.

- The general form of the PROJECT operation is
  $\pi_{\text{<attribute list>}}(R)$
  where π(pi) is the symbol used to represent the PROJECT operation, and <attribute list> is the desired sublist of attributes from the attributes of relation R.
- Again, notice that R is, in general, a *relational algebra expression* whose result is a relation, which in the simplest case is just the name of a database relation.
- The result of the PROJECT operation has only the attributes specified in <attribute list> *in the same order as they appear in the list*.
- Hence, its degree is equal to the number of attributes in <attribute list>.

- If the attribute list includes only nonkey attributes of R, duplicate tuples are likely to occur.
- The PROJECT operation *removes any duplicate tuples*, so the result of the PROJECT operation is a set of distinct tuples, and hence a valid relation.
- This is known as **duplicate elimination**.

- The number of tuples in a relation resulting from a PROJECT operation is always less than or equal to the number of tuples in R.
- If the projection list is a superkey of R—that is, it includes some key of R—the resulting relation has the *same number* of tuples as R.
- Moreover,

$$\pi_{\text{<list1>}}\ (\pi_{\text{<list2>}}(R)) = \pi_{\text{<list1>}}(R)$$

  as long as <list2> contains the attributes in <list1>; otherwise, the left-hand side is an incorrect expression.
- It is also noteworthy that commutativity *does not* hold on PROJECT.

- $\Pi_{\texttt{Lname, Fname, Salary}}$(EMPLOYEE)
- $\Pi_{\texttt{Sex, Salary}}$(EMPLOYEE)

- In SQL, the PROJECT attribute list is specified in the SELECT clause of a query.

- $\Pi_{\text{Sex, Salary}}(\text{EMPLOYEE})$
- ```
  SELECT DISTINCT Sex, Salary
  FROM   EMPLOYEE;
  ```
  ```

Sequences of Operations and the RENAME Operation

- For example, to retrieve the first name, last name, and salary of all employees who work in department number 5, we must apply a SELECT and a PROJECT operation.

- $\pi_{Fname,\ Lname,\ Salary}(\sigma_{Dno=5}(EMPLOYEE))$
- DEP5_EMPS $\leftarrow \sigma_{Dno=5}(EMPLOYEE)$
  RESULT $\leftarrow \pi_{Fname,\ Lname,\ Salary}(DEP5\_EMPS)$

- To *rename* the attributes in a relation, we simply list the new attribute names in parentheses.

- $\text{TEMP} \leftarrow \sigma_{\text{Dno=5}}(\text{EMPLOYEE})$
  $R(\text{First\_name, Last\_name, Salary}) \leftarrow \pi_{\text{Fname, Lname, Salary}}(\text{TEMP})$

- We can also define a formal **RENAME** operation—which can rename either the relation name or the attribute names, or both—as a unary operator.
- The general RENAME operation when applied to a relation R of degree n is denoted by any of the following three forms:

  $\rho_{S(B1, B2, ..., Bn)}(R)$ or
  $\rho_{S}(R)$ or
  $\rho_{(B1, B2, ..., Bn)}(R)$

  where the symbol ρ(rho) is used to denote the RENAME operator, S is the new relation name, and $B_1, B_2, ..., B_n$ are the new attribute names.

- In SQL, a single query typically represents a complex relational algebra expression.
- Renaming in SQL is accomplished by aliasing using **AS**.
- ```
  SELECT E.Fname AS First_name,
         E.Lname AS Last_name,
         E.Salary AS Salary
  FROM   EMPLOYEE AS E
  WHERE  E.Dno=5;
  ```
  ```