# Formal Query Languages: Relational Algebra

- Set Theory Operations
- Specific Relational Operations
- Write Queries in Relational Algebra



# Relational Algebra

- Operations on entire relations
  - Operands are (constant or variable) relations
  - Result is a relation
- Set theory operations:
  - Union, Intersection, Difference and Cartesian Product (product for short)
- Specific relational operations:
  - Selection, Projection, Join and Division
- Complete set of relational algebra operations:
  - Select, project, product, union and difference
- SQL is based on concepts from relational algebra



# Selection σ

- Unary operator Select, **σ**:

$$\sigma_{\text{selection-condition}}(\mathbf{r})$$

- E.g., $\sigma_{\text{Name= 'John'} \vee \text{Name = 'Susan'}}$ (STUDENT)
  - result = {t | t∈r and (t[Name] = 'John' or t[Name] = 'Susan')}
- *Selection condition* any logical expression on attributes of **r** involving any applicable comparison operator
  {=,<, ≤, >, ≥, ≠}



# Example of Selection

- $\sigma_{\text{Name='Bob'} \vee \text{ Major = 'Math'}}(S)$ = ?
- $\sigma_{\text{Name='Bob'} \wedge \text{ Major = 'Math'}}(S)$ = ?
- How can I get a copy of S?
- How can I get an empty copy of S?

Relation **S**

| SID | Name | Major |
|---|---|---|
| 1 | Bob | CS |
| 3 | Ann | CoE |
| 4 | Bob | Math |

## Projection

- Unary operator Project, Π:

  $\Pi_{\text{attribute-list}}(r)$

  - Attribute-list $\subseteq$ R

- E.g., $\Pi_{\text{Name, Major}}$ (STUDENT)
  - result = {t | t∈r and t[Name, Major]}

- What about $\Pi_{\text{SID, Major}}$ (S) = ?

Relation **S**

| SID | Name | Major |
|---|---|---|
| 1 | Bob | CS |
| 3 | Ann | CoE |
| 4 | Bob | Math |
| 5 | Bob | CS |



## Relational Algebra Expressions

- Query: *List the QPA of all students (SID) in CSD whose QPA is greater than 3.5*
- STUDENT (SID, FName, SName, Dept, Major, QPA)
- **Nesting** the operations

  $\Pi_{\text{SID, QPA}}(\sigma_{\text{Dept = 'CSD'} \wedge \text{QPA>3.5}}(\text{STUDENT}))$
- **Sequence** of operations

  $HS \leftarrow \sigma_{\text{Dept = 'CSD'} \wedge \text{QPA>3.5}}(\text{STUDENT})$

  $RESULT \leftarrow \Pi_{\text{SID, QPA}}(HS)$
- **Query tree**
  - *leaves* nodes are relations and *internal* nodes are operations



## Renaming Operator

- Renaming attributes of the result

  RSLT(StudentID, GPA) $\leftarrow \Pi_{\text{SID, QPA}}$ (HS)

- Change the name of Attributes (in general):

  $\rho(a1,a2,a3,..an)(r)$

- Example:

  $\rho$(StudentID, GPA) (

  $\Pi_{\text{SID, QPA}}(\sigma_{\text{Dept = 'CSD'} \wedge \text{QPA>3.5}}(\text{STUDENT}))$



## Properties of σ and π

- $\sigma_{cond1}(\sigma_{cond2}(R)) = \sigma_{cond2}(\sigma_{cond1}(R))$
- $\sigma_{cond1}(\sigma_{cond2}(R)) = \sigma_{cond2 \wedge cond1}(R)$

  $= \sigma_{cond1 \wedge cond2}(R)$
- $\Pi_{list1}(\Pi_{list2}(R)) = \Pi_{list1}(R)$ When?

## Efficient / Optimized Queries

- Reduce cost of computing (a.k.a, *time-complexity* )
  - Short-circuit (fast computing logical expressions)
  - Execute faster comparisons first
- Reduce memory needs (a.k.a., *space-complexity*)
  - Execute Selections with high *selectivity* (i.e., with more strict conditions) to reduce the size of intermediate tables.
  - Execute Projects as early as possible to reduce tuple size



## Selectivity

- Selectivity = The **ratio** of the number of records that satisfy a condition to the total number of records
- Let assume that Students
  - Female = 55% & Male 45%
  - CS majors = 5% & Non-CS majors = 95%
- Which is more efficient?
  - $\sigma_{\text{Major= 'Non-CS'} \wedge \text{Gender = 'Female'}}(\text{STUDENT})$
  - $\sigma_{\text{Gender = 'Female'} \wedge \text{Major= 'Non-CS'}}(\text{STUDENT})$
  - $\sigma_{\text{Major= 'CS'} \wedge \text{Gender = 'Female'}}(\text{STUDENT})$