CS422 Principles of Database Systems
More about Relational Algebra

Chengyu Sun
California State University, Los Angeles

Why Study Relational Algebra?

•... when we already know SQL?
•Because we want to know
  •What basic operations are needed for a query language like SQL?
  •Design
  •Implementation
  •What the capabilities and limitations of such a language?
  •How to we decide whether two queries are equivalent
  •Query optimization

Notations

<table>
<thead>
<tr>
<th>OPERATION</th>
<th>SYMBOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNION</td>
<td>∪</td>
</tr>
<tr>
<td>INTERSECTION</td>
<td>∩</td>
</tr>
<tr>
<td>DIFFERENCE</td>
<td>−</td>
</tr>
<tr>
<td>SELECT</td>
<td>σ&lt;sub&gt;i&lt;/sub&gt;</td>
</tr>
<tr>
<td>PROJECTION</td>
<td>π&lt;sub&gt;i&lt;/sub&gt;</td>
</tr>
<tr>
<td>JOIN</td>
<td>⋈&lt;sub&gt;i&lt;/sub&gt;</td>
</tr>
<tr>
<td>RENAME</td>
<td>ρ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OPERATION</th>
<th>SYMBOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTA</td>
<td>δ</td>
</tr>
<tr>
<td>TAU</td>
<td>τ</td>
</tr>
<tr>
<td>GAMMA</td>
<td>γ</td>
</tr>
<tr>
<td>PROJECTION</td>
<td>π</td>
</tr>
<tr>
<td>OUTERJOIN</td>
<td>⊗</td>
</tr>
<tr>
<td>LEFTJOIN</td>
<td>⋈</td>
</tr>
<tr>
<td>RIGHTJOIN</td>
<td>⊕</td>
</tr>
<tr>
<td>SUM, AVG, COUNT, MIN, MAX</td>
<td>Σ, ∑</td>
</tr>
</tbody>
</table>

Constraints on Relations

•Constraints
  •not null, unique, primary key, references (foreign key) ...
•Express constraints using relational algebra
  •R = ∅
  •R ⊆ S

Two Observations

•The two ways of expressing constraints in relational algebra are equivalent
  •R = ∅ ⊨ R ⊆ ∅
  •R ⊆ S ⊨ R - S = ∅
•R and S could be sets or bags

Example

Products (ProdID, Description, Price)
Orders (OrderID, ProdID, Quantity)
Referential Integrity Constraints

- A value of ProdID in the Orders relation must match a value of ProdID in the Products relation

Functional Dependency

- \{ ProdID \} \rightarrow \{ Description, Price \}
- \{ OrderID, ProdID \} \rightarrow \{ Quantity \}

Other Constraints

- Not NULL
  - ProdID in Products cannot be null
- Unique
  - ProdID in Products has to be unique
- Enumeration
  - Gender attribute in a relation R has to be either "Male" or "Female"

Exercises

- Core Relation Algebra
  - 5.2.1, 5.2.4
- Extended Operators
  - 5.4.1, 5.4.3
- Constraints
  - 5.5.1, 5.5.2

Computer Products DB

- Product ( maker, model, type )
- PC ( model, speed, ram, hd, rd, price )
- Laptop ( model, speed, ram, hd, screen, price )
- Printer ( model, color, type, price )

  model number is assumed to be unique over all manufacturers and product types.

WWII Capital Ships DB

- Classes ( class, type, county, numGuns, bore, displacement )
- Ships ( name, class, launched )
- Battles ( name, date )
- Outcomes ( ship, battle, result )