Announcements

- WQ3 is due Sunday 11pm
- Azure codes will be sent out Wed/Thu
- Don’t miss section tomorrow
  – will go through Azure setup and basic use
- HW3 will be posted by Thu night
  – due on Tuesday, 4/25 (in 13 days)

Where We Are

• Motivation for using a DBMS for managing data
• SQL:
  – Declaring the schema for our data (CREATE TABLE)
  – Inserting data one row at a time or in bulk (INSERT/import)
  – Modifying the schema and updating the data (ALTER/UPDATE)
  – Querying the data (SELECT)
• Next step: More knowledge of how DBMSs work
  – Client-server architecture
  – Relational algebra and query execution

Query Evaluation Steps

1. Parse & Check Query
   - Translate query string into internal representation
2. Decide how best to answer query: query optimization
3. Logical plan -> physical plan
4. Query Execution
5. Return Results

Query Evaluation

The WHAT and the HOW

• SQL = WHAT we want to get from the data
• Relational Algebra = HOW to get the data we want

• Move from WHAT to HOW is query optimization
  – SQL ➔ Relational Algebra ➔ Physical Plan
  – Relational Algebra = Logical Plan

Relational Algebra
Sets v.s. Bags

- Sets: \{a,b,c\}, \{a,d,e,f\}, \{\}\ldots
- Bags: \{a, a, b, c\}, \{b, b, b, b, b\}, \ldots

Relational Algebra has two semantics:
- Set semantics = standard Relational Algebra
- Bag semantics = extended Relational Algebra

DB systems implement bag semantics (Why?)

Relational Algebra Operators

- Union \(\cup\), intersection \(\cap\), difference \(-\)
- Selection \(\sigma\)
- Projection \(\pi(\Pi)\)
- Cartesian product \(\times\), join \(\Join\)
- Rename \(\rho\)
- Duplicate elimination \(\delta\)
- Grouping and aggregation \(\gamma\)
- Sorting \(\tau\)

Union and Difference

\[ R1 \cup R2 \]
\[ R1 - R2 \]

What do they mean over bags?

What about Intersection?

- Derived operator using minus
  \[ R1 \cap R2 = R1 - (R1 - R2) \]
- Derived using join (will explain later)
  \[ R1 \cap R2 = R1 \Join R2 \]

Selection

- Returns all tuples which satisfy a condition
  \[ \sigma_c(R) \]
- Examples
  - \(\sigma_{\text{Salary} > 40000}(\text{Employee})\)
  - \(\sigma_{\text{Salary} = \text{Other}}(\text{Employee})\)
- The condition \(c\) can be =, \(<\), \(\leq\), \(>\), \(\geq\), \(<>\) combined with AND, OR, NOT

Employee

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234545</td>
<td>John</td>
<td>20000</td>
</tr>
<tr>
<td>5423341</td>
<td>Smith</td>
<td>60000</td>
</tr>
<tr>
<td>4352342</td>
<td>Fred</td>
<td>50000</td>
</tr>
</tbody>
</table>

\(\sigma_{\text{Salary} > 60000}(\text{Employee})\)

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>5423341</td>
<td>Smith</td>
<td>60000</td>
</tr>
<tr>
<td>4352342</td>
<td>Fred</td>
<td>50000</td>
</tr>
</tbody>
</table>
Projection

• Eliminates columns

\[ \pi_{A_1, \ldots, A_n}(R) \]

• Example: project social-security number and names:
  - \( \pi_{\text{SSN}, \text{Name}}(\text{Employee}) \)
  - Answer(\text{SSN}, \text{Name})

Different semantics over sets or bags! Why?

Composing RA Operators

Patient

\[
\begin{array}{ccc}
\text{no} & \text{name} & \text{zip} & \text{disease} \\
1 & p1 & 98125 & \text{flu} \\
2 & p2 & 98125 & \text{heart} \\
3 & p3 & 98120 & \text{lung} \\
4 & p4 & 98120 & \text{heart} \\
\end{array}
\]

Patient

\[\pi_{\text{disease}=\text{heart}}(\text{Patient})\]

\[
\begin{array}{ccc}
\text{no} & \text{name} & \text{zip} & \text{disease} \\
2 & p2 & 98125 & \text{heart} \\
4 & p4 & 98120 & \text{heart} \\
\end{array}
\]

Patient

\[\pi_{\text{zip}=\text{disease}}(\text{Patient})\]

\[
\begin{array}{ccc}
\text{zip} & \text{disease} \\
98125 & \text{flu} \\
98125 & \text{heart} \\
98120 & \text{heart} \\
\end{array}
\]

Cartesian Product

• Each tuple in R1 with each tuple in R2

\[ R_1 \times R_2 \]

• Rare in practice; mainly used to express joins

Cross-Product Example

Employee

\[
\begin{array}{ccc}
\text{Name} & \text{SSN} \\
\text{John} & 999999999 \\
\text{Tony} & 777777777 \\
\end{array}
\]

Dependent

\[
\begin{array}{ccc}
\text{Name} & \text{EmpSSN} & \text{DepName} \\
\text{Emily} & 999999999 \\
\text{Joe} & 777777777 \\
\end{array}
\]

Employee \times Dependent

\[
\begin{array}{ccc}
\text{Name} & \text{SSN} & \text{EmpSSN} & \text{DepName} \\
\text{John} & 999999999 & 999999999 & \text{Emily} \\
\text{Tony} & 777777777 & 999999999 & \text{Emily} \\
\end{array}
\]

Renaming

• Changes the schema, not the instance

\[ \rho_{B_1, \ldots, B_n}(R) \]

• Example:
  - \( \rho_{\text{Name}, \text{SSN}}(\text{Employee}) \rightarrow \text{Answer}(\text{N, S}) \)

Not really used by systems, but needed on paper
**Natural Join**

\[ R1 \bowtie R2 \]

- Meaning: \( R1 \bowtie R2 = \pi_A (\sigma_{R1 \bowtie R2} (R1 \times R2)) \)
- Where:
  - Selection \( \sigma \) checks equality of all common attributes (attributes with same names)
  - Projection \( \pi \) eliminates duplicate common attributes

**Natural Join Example**

\[
\begin{array}{c|c}
A & B \\
X & Y \\
X & Z \\
Y & Z \\
Z & V \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c}
R & S & B & C \\
\hline
X & Z & U \\
X & Z & V \\
Y & Z & U \\
Y & Z & V \\
Z & V & W \\
\end{array}
\]

\( R \bowtie S = \rho_{ABC} (\sigma_{R.B = S.B} (R \times S)) \)

**Natural Join Example 2**

<table>
<thead>
<tr>
<th>AnonPatient P</th>
<th>Voters V</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>zip</td>
</tr>
<tr>
<td>54</td>
<td>98125</td>
</tr>
<tr>
<td>20</td>
<td>98120</td>
</tr>
</tbody>
</table>

\( P \bowtie V \)

<table>
<thead>
<tr>
<th>age</th>
<th>zip</th>
<th>disease</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>54</td>
<td>98125</td>
<td>heart</td>
<td>p1</td>
</tr>
<tr>
<td>20</td>
<td>98120</td>
<td>flu</td>
<td>p2</td>
</tr>
</tbody>
</table>

**Natural Join**

- Given schemas \( R(A, B, C, D) \), \( S(A, C, E) \), what is the schema of \( R \bowtie S \)?
- Given \( R(A, B, C) \), \( S(D, E) \), what is \( R \bowtie S \)?
- Given \( R(A, B) \), \( S(A, B) \), what is \( R \bowtie S \)?

**Theta Join**

- A join that involves a predicate

\[ R1 \bowtie_\theta R2 = \sigma_{\theta} (R1 \times R2) \]

- Here \( \theta \) can be any condition
- For our voters/patients example:

\[ P \bowtie Q zip = V.zip \text{ and } Page \geq V.age - 1 \text{ and } Page \leq V.age + 1 \]

**Equijoin**

- A theta join where \( \theta \) is an equality predicate
- By far the most used variant of join in practice
Equijoin Example

AnonPatient P

<table>
<thead>
<tr>
<th>age</th>
<th>zip</th>
<th>disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>54</td>
<td>98125</td>
<td>heart</td>
</tr>
<tr>
<td>20</td>
<td>98120</td>
<td>flu</td>
</tr>
</tbody>
</table>

Voters V

<table>
<thead>
<tr>
<th>name</th>
<th>age</th>
<th>zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>54</td>
<td>98125</td>
</tr>
<tr>
<td>p2</td>
<td>20</td>
<td>98120</td>
</tr>
</tbody>
</table>

P \Join_{P.age = V.age} V

<table>
<thead>
<tr>
<th>P.age</th>
<th>P.zip</th>
<th>P.disease</th>
<th>P.name</th>
<th>V.zip</th>
<th>V.age</th>
</tr>
</thead>
<tbody>
<tr>
<td>54</td>
<td>98125</td>
<td>heart</td>
<td>p1</td>
<td>54</td>
<td>98125</td>
</tr>
<tr>
<td>20</td>
<td>98120</td>
<td>flu</td>
<td>p2</td>
<td>20</td>
<td>98120</td>
</tr>
</tbody>
</table>

Join Summary

- **Theta-join**: $R \bowtie_S S = \sigma_\theta(R \times S)$
  - Join of $R$ and $S$ with a join condition $\theta$
  - Cross-product followed by selection $\sigma_\theta$
- **Equijoin**: $R \bowtie_S S = \pi_A (\sigma_\theta(R \times S))$
  - Join condition $\theta$ consists only of equalities
- **Natural join**: $R \bowland_S S = \pi_A (\sigma_\theta(R \times S))$
  - Equijoin
  - Equality on all fields with same name in $R$ and in $S$
  - Projection $\pi_A$ drops all redundant attributes

So Which Join Is It?

When we write $R \bowland S$ we usually mean an equijoin, but we often omit the equality predicate when it is clear from the context.

More Joins

- **Outer join**
  - Include tuples with no matches in the output
  - Use NULL values for missing attributes
  - Does not eliminate duplicate columns

- **Variants**
  - Left outer join
  - Right outer join
  - Full outer join

More Examples

- **Supplier**
  - $\sigma_{\text{psize} > 10}(\text{Supply} \bowland \text{Part})$

- **Part**
  - $\sigma_{\text{pcolor} = 'red' \lor \text{psize} > 10}(\text{Part})$

- **Name of supplier of parts with size greater than 10**
  - $\pi_{\text{sname}}(\text{Supplier} \bowland \text{Supply} \bowland \text{Part})$

- **Name of supplier of red parts or parts with size greater than 10**
  - $\pi_{\text{sname}}(\text{Supplier} \bowland \text{Supply} \bowland \text{Part} \bowland \text{Part})$