Relational Database Manipulation
Robert C. Nickerson
ISYS 464 – Spring 2003
Topic 08

Basic Relational Database Query Operations

• Three basic relational database query operations:
  – Projection: retrieve selected attributes from a relation
  – Selection (Restriction): retrieve selected tuples from a relation
  – Join: combine two or more relations

• Note: All relational database query operations take one or more relations as input and produce one relation as output

Basic Relational Database Update Operations

• Three basic relational database update operations:
  – Add: insert one or more new tuples in a relation
  – Delete: remove one or more existing tuples from a relation
  – Change: modify certain data in one or more tuples of a relation

• Note: All relational database update operations take one relation as input and produce a modified form of that relation as output

Projection

• The operation of retrieving certain attributes from a relation
  • Can project one or more attributes
  • Example: Project the Student Name and Major attribute from the Student relation

Student Course Database
Student (Student Number, Student Name, Major)
Course (Course Number, Course Name, Day, Time)
Student Course (Student Number, Course Number)

Selection (Restriction)

• The operation of retrieving certain tuples from a relation based on a condition
  • Selection can result in one or more tuples, or no tuples if no tuples in the relation satisfy the condition
  • Example: Select all tuples from the Course relation where the course is taught on TTh

Copyright (c) 2003 by Robert C. Nickerson. All rights reserved. Not for general distribution.
Join

- The operation of combining the tuples of two relations based on a condition
- Example: Join the Student relation and the Student Course relation based on the Student Number in the Student relation equaling the Student Number in the Student Course relation

Example: Join

<table>
<thead>
<tr>
<th>Student Number</th>
<th>Student Name</th>
<th>Course Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>Joe</td>
<td>ISYS 365</td>
</tr>
<tr>
<td>234</td>
<td>Fred</td>
<td>ISYS 365</td>
</tr>
<tr>
<td>345</td>
<td>Mary</td>
<td>ISYS 464</td>
</tr>
<tr>
<td>456</td>
<td>Sue</td>
<td>MGMT 405</td>
</tr>
<tr>
<td>567</td>
<td>Lee</td>
<td>ISYS 464</td>
</tr>
</tbody>
</table>

Example: Join

<table>
<thead>
<tr>
<th>Student Course</th>
<th>Student Number</th>
<th>Course Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>ISYS 365</td>
<td></td>
</tr>
<tr>
<td>234</td>
<td>ISYS 365</td>
<td></td>
</tr>
<tr>
<td>345</td>
<td>ISYS 464</td>
<td></td>
</tr>
<tr>
<td>456</td>
<td>MGMT 405</td>
<td></td>
</tr>
<tr>
<td>567</td>
<td>ISYS 464</td>
<td></td>
</tr>
<tr>
<td>678</td>
<td>ISYS 365</td>
<td></td>
</tr>
</tbody>
</table>

Join Terminology

- **Join condition**: the condition that indicates under what circumstances tuples in the two relations are to be combined
- **Note**: We can have a join involving any of the relational operators: =, <>, >, >=, <, <=
- **Equijoin**: a join involving an equal join condition
- **Note**: Only with an equijoin are we guaranteed to have two attributes that are identical in the result of the join
- **Natural join**: an equijoin in which one of the duplicate attributes is automatically eliminated

Join Use

- **Join** is the operation we use to utilize relationships in relational databases:
  - By joining two relations based on the primary key of one relation equaling the corresponding foreign key of another relation, we match related tuples from the two relations
  - **Join with different relationships**:
    - **One-to-one**: Each tuple of one relation will match zero or one tuples of the other relation
    - **One-to-many**: Each tuple of the parent relation will match zero, one, or more tuples of the child relation
    - **Many-to-many**: Requires joining three relations, the two related relations and the intersection relation. Each tuple of one related relation is matched with related tuples in the other related relation, and vice versa, through the intersection relation.
  - **Note**: Join can result in no tuples if condition is not met

Inner and Outer Joins

- **Inner join**: a join in which tuples in the two joined relations that do not satisfy the join condition are not included in the result
  - **Note**: An inner join is assumed unless otherwise stated
- **Example**: Previous example was an inner join
- **Outer join** (full outer join): a join in which tuples in the two joined relations that do not satisfy the join condition are included in the result
  - **Example**: Outer join of Student and Student Course relations based on the Student Number in the Student relation equaling the Student Number in the Student Course relation
Example: Outer Join

Output

<table>
<thead>
<tr>
<th>Student Number</th>
<th>Student Name</th>
<th>Major</th>
<th>Student Number</th>
<th>Course Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>Joe</td>
<td>IS</td>
<td>123</td>
<td>FIN 350</td>
</tr>
<tr>
<td>123</td>
<td>Joe</td>
<td>IS</td>
<td>123</td>
<td>MGMT 405</td>
</tr>
<tr>
<td>234</td>
<td>Fred</td>
<td>IS</td>
<td>234</td>
<td>FIN 350</td>
</tr>
<tr>
<td>234</td>
<td>Fred</td>
<td>IS</td>
<td>234</td>
<td>MGMT 405</td>
</tr>
<tr>
<td>345</td>
<td>Mary</td>
<td>Acct</td>
<td>345</td>
<td>ISYS 464</td>
</tr>
<tr>
<td>346</td>
<td>Lee</td>
<td>Acct</td>
<td>346</td>
<td>ISYS 365</td>
</tr>
<tr>
<td>347</td>
<td>Lee</td>
<td>Acct</td>
<td>347</td>
<td>MGMT 405</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>ISYS 365</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>678</td>
<td>MGMT 405</td>
</tr>
</tbody>
</table>

Note NULLs for attributes of unmatched tuples

Left and Right Outer Join

- **Left outer join**: a join in which tuples in the relation named first (on the left) that do not satisfy the join condition are included in the result
- **Right outer join**: a join in which tuples in the relation named second (on the right) that do not satisfy the join condition are included in the result

Example: Left Outer Join

Output

<table>
<thead>
<tr>
<th>Student Number</th>
<th>Student Name</th>
<th>Major</th>
<th>Student Number</th>
<th>Course Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>Joe</td>
<td>IS</td>
<td>123</td>
<td>FIN 350</td>
</tr>
<tr>
<td>123</td>
<td>Joe</td>
<td>IS</td>
<td>123</td>
<td>MGMT 405</td>
</tr>
<tr>
<td>234</td>
<td>Fred</td>
<td>IS</td>
<td>234</td>
<td>FIN 350</td>
</tr>
<tr>
<td>234</td>
<td>Fred</td>
<td>IS</td>
<td>234</td>
<td>MGMT 405</td>
</tr>
<tr>
<td>345</td>
<td>Mary</td>
<td>Acct</td>
<td>345</td>
<td>ISYS 464</td>
</tr>
<tr>
<td>346</td>
<td>Lee</td>
<td>Acct</td>
<td>346</td>
<td>ISYS 365</td>
</tr>
<tr>
<td>347</td>
<td>Lee</td>
<td>Acct</td>
<td>347</td>
<td>MGMT 405</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>ISYS 365</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>678</td>
<td>MGMT 405</td>
</tr>
</tbody>
</table>

Example: Right Outer Join

Output

<table>
<thead>
<tr>
<th>Student Number</th>
<th>Student Name</th>
<th>Major</th>
<th>Student Number</th>
<th>Course Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>Joe</td>
<td>IS</td>
<td>123</td>
<td>FIN 350</td>
</tr>
<tr>
<td>123</td>
<td>Joe</td>
<td>IS</td>
<td>123</td>
<td>MGMT 405</td>
</tr>
<tr>
<td>234</td>
<td>Fred</td>
<td>IS</td>
<td>234</td>
<td>FIN 350</td>
</tr>
<tr>
<td>234</td>
<td>Fred</td>
<td>IS</td>
<td>234</td>
<td>MGMT 405</td>
</tr>
<tr>
<td>345</td>
<td>Mary</td>
<td>Acct</td>
<td>345</td>
<td>ISYS 464</td>
</tr>
<tr>
<td>346</td>
<td>Lee</td>
<td>Acct</td>
<td>346</td>
<td>ISYS 365</td>
</tr>
<tr>
<td>347</td>
<td>Lee</td>
<td>Acct</td>
<td>347</td>
<td>MGMT 405</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>ISYS 365</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>678</td>
<td>MGMT 405</td>
</tr>
</tbody>
</table>

Relational Database Languages

- Three main sub-languages in a relational database language
  - Data Manipulation Language (DML) – used to query and update the data in a database
  - Data Definition Language (DDL) – used to define the characteristics of a database
  - Data Control Language (DCL) – used to control user access to a database

Data Manipulation Language (DML)

- A language used to query and update the data in a database
- A relational database is manipulated by a relational DML
- Two main types of relational DMLs:
  - Relational algebra
  - Relational calculus

Copyright (c) 2003 by Robert C. Nickerson. All rights reserved. Not for general distribution.
Relational Algebra

- Procedural (like algebra: \( x^2 + 2x + 3 \))
- States how to obtain result by giving the sequence of operations
- Example using hypothetical relational algebra: What are the names of all students taking ISYS 263?
  
  \[
  \text{Student\_Course WHERE Course\_Number = 'ISYS 263' JOIN (Student\_Course.Student\_Number = Student.Student\_Number)}
  \]
  
  \text{Student [Student\_Name]}

- Processing performed left to right through the command

Relational Calculus

- Non-procedural (like calculus: \( \int x^2 \, dx \))
- States what is wanted, not how to obtain it
- Query by Example (QBE) is a type of relational calculus
- Example using QBE: What are the names of all students taking ISYS 263?

Order of processing is not given
Order determined by software
QBE is sometimes called a graphical language

Hybrid Relational DML

- Combines elements of relational algebra and relational calculus
- Some procedural and some non-procedural characteristics
- States what is wanted but may state some sequencing of operations
- Example using SQL: What are the names of all students taking ISYS 263?

\[
\begin{align*}
\text{SELECT Student\_Name} \\
\text{FROM Student, Student\_Course} \\
\text{WHERE Student.Student\_Number = Student\_Course.Student\_Number} \\
\text{AND Course\_Number = 'ISYS 263'}
\end{align*}
\]

- SQL is sometimes called a transform-oriented language

Data Definition Language (DDL)

- A language used to define the characteristics of a database
- DDL permits:
  - Definition of relations in the conceptual view of the database
  - Definition of relations in each external view of the database
  - Definition of structures used to enhance performance (e.g., indexes)

Data Control Language (DCL)

- A language used to control user access to the database
- DCL permits:
  - Giving of privileges to users (e.g., access, update privileges)
  - Removing privileges from users