Database Management Systems

Course 236363

Tutorial 5:
Relational Calculus
Outline

• Relational Calculus
  – Signatures and structures
  – Library database
• Relational Calculus queries
• Domain Independence
  – Semantics
  – Definition
  – Examples
• Queries normal forms
Relational Calculus

• We can define queries with Relational Calculus

• In relational calculus
  – We have signature that contains
    • Relation symbols
    • Functions symbols
    • Constants
  – Structure that gives interpretation to a signature
    • Specifies the domain
    • The tuples in the relations
    • The functions
    • The constants
Example

- The signature
  \[ \langle R_1(\circ, \circ, \circ, \circ, \circ, \circ), R_2(\circ, \circ, \circ), R_3(\circ, \circ, \circ), R_4(\circ, \circ, \circ, \circ) \rangle \]
  - Has a 6-ary relation (R1), two 3-ary relations (R2, R3), and a 4-ary relation (R4)

- The following structure gives interpretation to the above signature
  \[ \langle \text{Str}, r_{\text{books}}, r_{\text{customers}}, r_{\text{ordered}}, r_{\text{borrowed}} \rangle \]
  - Str is the set of all finite strings
  - Note that the names of the attributes are not part of the structure
The relations

- \( <\text{Str}, r_{\text{books}}, r_{\text{customers}}, r_{\text{ordered}}, r_{\text{borrowed}} > \)

<table>
<thead>
<tr>
<th>ISBN</th>
<th>Title</th>
<th>Year</th>
<th>Issue</th>
<th>Pages</th>
<th>Faculty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>Database Systems</td>
<td>1998</td>
<td>7</td>
<td>348</td>
<td>CS</td>
</tr>
<tr>
<td>1112</td>
<td>Database Systems</td>
<td>1998</td>
<td>14</td>
<td>348</td>
<td>CS</td>
</tr>
<tr>
<td>1113</td>
<td>Database Systems</td>
<td>2001</td>
<td>7</td>
<td>424</td>
<td>CS</td>
</tr>
<tr>
<td>2222</td>
<td>Database And Knowledge</td>
<td>1998</td>
<td>1</td>
<td>390</td>
<td>CS</td>
</tr>
<tr>
<td>2223</td>
<td>Database And Knowledge</td>
<td>1998</td>
<td>7</td>
<td>390</td>
<td>EE</td>
</tr>
</tbody>
</table>
The relations

- \(<\text{Str}, r_{\text{books}}, r_{\text{customers}}, r_{\text{ordered}}, r_{\text{borrowed}}>\)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12345</td>
<td>Moshe Cohen</td>
<td>CS</td>
</tr>
<tr>
<td>23456</td>
<td>Avi Barak</td>
<td>EE</td>
</tr>
<tr>
<td>34567</td>
<td>Avi Barak</td>
<td>MED</td>
</tr>
<tr>
<td>45678</td>
<td>Lior Edri</td>
<td>EE</td>
</tr>
<tr>
<td>56789</td>
<td>Moshe Cohen</td>
<td>EE</td>
</tr>
<tr>
<td>67890</td>
<td>Moshe Cohen</td>
<td>EE</td>
</tr>
</tbody>
</table>
The relations

- \(<Str, r_{books}, r_{customers}, r_{ordered} \cdot r_{borrowed}>\)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12345</td>
<td>1111</td>
<td>14-Oct-2002</td>
</tr>
<tr>
<td>45678</td>
<td>1112</td>
<td>24-Oct-2002</td>
</tr>
<tr>
<td>12345</td>
<td>1113</td>
<td>30-Oct-2002</td>
</tr>
<tr>
<td>45678</td>
<td>2222</td>
<td>12-Oct-2002</td>
</tr>
</tbody>
</table>
The relations

- \(<\text{Str}, r_{\text{books}}, r_{\text{customers}}, r_{\text{ordered}}, r_{\text{borrowed}})>\n
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
• We assume that all of the elements in the domain appear as constants
  – each element can appear in a formula
Outline

• Relational Calculus
  – Signatures and structures
  – Library database

• Relational Calculus queries

• Domain Independence
  – Semantics
  – Definition
  – Examples

• Queries normal forms
• A reminder:
  – Customers(custId, custName, faculty)
• What does the following query return?
\{ n : \exists i, f \left[ \text{Customers}(i, n, f) \land \exists i', f'( \text{Customers}(i', n, f') \land R\neq (i, i')) \right] \}

• The “ambiguous” names
Relational Calculus Queries

• A reminder:
  – Books(id, year, name, time, pages, faculty)

• What does the following query return?
{ n : \exists i, y, t, p, f
  
  [Books(i, y, n, t, p, f) \land
   \exists n', t', p', f' Books("1112", y, n', t', p', f') ] }

• The names of the books that were published in the same year as book “1112”
Relational Calculus Queries

• A reminder:
  – Books(id, year, name, time, pages, faculty)
• What does the following query return?

\[
\{ n : \exists i, y, t, p, f \quad \exists i', y', n', t', p' \quad [\text{Books}(i, y, n, t, p, f) \land \\
\quad \forall i', y', n', t', p' \quad \text{Books}(i', y', n', t', p', "CS") \rightarrow R_<(p', p)] \}
\]

• The names of the books that have more pages than each of the books in the CS department
Outline

• Relational Calculus
  – Signatures and structures
  – Library database
• Relational Calculus queries
• Domain Independence
  – Semantics
  – Definition
  – Examples
• Queries normal forms
In databases, the domain is the set of finite strings.

The semantics of a query is defined by assigning values to the variables.

The values are elements in the domain.

However, we want to phrase queries that are independent of the domain as will be defined shortly.
Semantics

• Given the formula $\forall x \, \psi(x)$
  – In fact, this is a sentence since there are no free variables
  – It is evaluated true if
    • For every element $x$ in the domain, it holds that $\psi(x)$
    • Similarly, $\exists x \, \psi(x)$ is evaluated true if there exists an element $x$ such that $\psi(x)$
  – Therefore, if two structures have the same relations and functions but different domains they are not equivalent.
Example

- Let
  - \( <U(\circ)> \) be a signature
    - Has only a single unary relation
  - \( \forall x \) \( U(x) \) be a formula (a sentence)
  - There are different structures over this signature
    - Characterized by their domain and by the relation \( U \)
- The semantics of this formula depends on the structure
  - \( <D=\{1,2,3\}, U=\{(1),(2),(3)\}> \rightarrow true \)
  - \( <D=\{1,2,3,4\}, U=\{(1),(2),(3)\}> \rightarrow false \)
Domain Independence

• We are interested in queries that are independent of the domain
• That is,
  – Let us denote the set of values for which the formula $\psi(x_1,\ldots,x_n)$ is evaluated true in a structure $M$ by $I_M$:
    - $I_M = \{(x_1,\ldots,x_n) : \psi(x_1,\ldots,x_n) \}$
• A formula $\psi(x_1,\ldots,x_n)$ over a signature $\tau$ is domain independent if for every structure $M$ over $\tau$ it holds that
  – If $M'$ is a structure over $\tau$ that is different than $M$ only in its domain, it holds that $I_M = I_{M'}$
• Example of formula which is not domain independent:
  – ∀x U(x)
  – Can you find two equivalent structures that differ only in their domains such that for one this formula is evaluated true and for the other false?

• Example of formula which is domain independent:
  – ∃x [S(x) ∧ ¬R(x)]
• Relational Calculus
  – Signatures and structures
  – Library database
• Relational Calculus queries
• Domain Independence
  – Semantics
  – Definition
  – Examples
• Queries normal forms
Formulas and Quantifiers

• We can move quantifiers to the beginning of the formula:
  • $\forall x \ (\phi_1 \land \phi_2) \equiv (\forall x \ \phi_1 \land \forall x \ \phi_2)$
    – However $\forall x \ (\phi_1 \lor \phi_2)$ is not equivalent to $\forall x \ \phi_1 \lor \forall x \ \phi_2$
    – For instance,
      • $\phi_1(x)$ iff $x$ is even, $\phi_2(x)$ iff $x$ is odd
  • $\exists x \ (\phi_1 \lor \phi_2) \equiv (\exists x \ \phi_1 \lor \exists x \ \phi_2)$
    – However $\exists x \ (\phi_1 \land \phi_2)$ is not equivalent to $(\exists x \ \phi_1 \land \exists x \ \phi_2)$
    – Same example as above
• If $x$ is not free variable in $\varphi_2$:
  - $\forall x \ (\varphi_1 \land \varphi_2) \equiv (\forall x \ \varphi_1) \land \varphi_2$

• Similarly for $\exists$ and $\lor$:
  - $\forall x \ (\varphi_1 \lor \varphi_2) \equiv (\forall x \ \varphi_1) \lor \varphi_2$
  - $\exists x \ (\varphi_1 \lor \varphi_2) \equiv (\exists x \ \varphi_1) \lor \varphi_2$
  - $\exists x \ (\varphi_1 \land \varphi_2) \equiv (\exists x \ \varphi_1) \land \varphi_2$
Formulas and Quantifiers

• De-Morgan rules:
  - \( \neg(\varphi_1 \lor \varphi_2) \equiv (\neg \varphi_1) \land (\neg \varphi_2) \)
  - \( \neg(\varphi_1 \land \varphi_2) \equiv (\neg \varphi_1) \lor (\neg \varphi_2) \)

• De-Morgan rules for quantifiers:
  - \( \neg (\forall x \varphi) \equiv \exists x \neg \varphi \)
  - \( \neg (\exists x \varphi) \equiv \forall x \neg \varphi \)

• Implication:
  - \( \varphi \rightarrow \mu \equiv \neg \varphi \lor \mu \)

• Changing variables names
Examples

• Write an equivalent formula in PNF (all of the quantifiers are in the beginning):
  \[ \forall x (R1(x,y) \rightarrow \exists z R2(y,z)) \land \exists x R2(x,y) \]

  \[ \forall x (\neg R1(x,y) \lor \exists z R2(y,z)) \land \exists x R2(x,y) \]
  \[ \forall x \exists z (\neg R1(x,y) \lor R2(y,z)) \land \exists x R2(x,y) \]
  \[ \forall x \exists z (\neg R1(x,y) \lor R2(y,z)) \land \exists w R2(w,y) \]
  \[ \forall x \exists z \exists w [(\neg R1(x,y) \lor R2(y,z)) \land R2(w,y)] \]