QBE

• Query-By-Example
  – provides a visual interface for queries and updates
  – a version supported by Microsoft Access (Graphical QBE)

• Examples: movie database queries
  – “Find the titles of currently playing movies”
    schedule | theater | title
    P.
    • P. : “print value”
  – “Find the titles of all movies by Berto”
    movie | title | director | actor
    P.  | Berto

QBE (2)

  – “Find the titles and directors of all currently playing movies”
    movie | title | director | actor
    _t  | _d
    schedule | theater | title
    _t
    result | title | director
    _t | _d

• Note:
  – answer table explicitly specified
  – underscore _x means _x can take any value, like a variable
  – I. means insert
QBE (3)

- “Find all actors playing in every movie by Berto”
  - requires multi-stage query, creating intermediate answers
  - analog of nested queries in SQL
- I stage:
  ```
  schedule | title  | director | actor | bad-actor | actor
  -------- | ------ | -------- |------ | --------- |------
  _a       | _t     | Berto    | _a   | _a        |
  ¬        | _t     | _a       |
  ```
- Semantics of ¬
  - for _t and _a fixed, satisfying positive part of pattern, there is no tuple occurring with _t and _a as in the negated tuple

QBE (4)

- II stage
  - (complement of temp computed in stage I)
  ```
  movie   | title  | director | actor
  ------- | ------ | -------- |------
  _a      |
  bad-actor | actor
  ¬        | _a    |
  result  | actor |
  _a      |
  ```
Find the drinkers who frequent some bars serving Coors

<table>
<thead>
<tr>
<th>frequents</th>
<th>drinker</th>
<th>bar</th>
<th>serves</th>
<th>bar</th>
<th>beer</th>
<th>likes</th>
<th>drinker</th>
<th>beer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>answer</td>
<td>drinker</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Find the drinkers who frequent at least one bar serving a beer they like

<table>
<thead>
<tr>
<th>frequents</th>
<th>drinker</th>
<th>bar</th>
<th>serves</th>
<th>bar</th>
<th>beer</th>
<th>likes</th>
<th>drinker</th>
<th>beer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>answer</td>
<td>drinker</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Find the drinkers who frequent ONLY bars serving a beer they like

<table>
<thead>
<tr>
<th>frequents</th>
<th>drinker</th>
<th>bar</th>
<th>serves</th>
<th>bar</th>
<th>beer</th>
<th>likes</th>
<th>drinker beer</th>
<th>answer</th>
<th>drinker</th>
</tr>
</thead>
</table>

 Updates in QBE

- **Deletions:** similar to inserts
  - D.
  - “Delete all movies by Berto”:

<table>
<thead>
<tr>
<th>movie</th>
<th>title</th>
<th>director</th>
<th>actor</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.</td>
<td></td>
<td></td>
<td>Berto</td>
</tr>
</tbody>
</table>

  - “Delete all movies by directors who are also actors”:

<table>
<thead>
<tr>
<th>movie</th>
<th>title</th>
<th>director</th>
<th>actor</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.</td>
<td></td>
<td>_d</td>
<td>_d</td>
</tr>
</tbody>
</table>
Updates in QBE (2)

- Updates: using **primary key** attributes
  - primary keys are explicitly declared
  
  - “Sally gets a 5% salary raise”

<table>
<thead>
<tr>
<th>employee</th>
<th>name</th>
<th>salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>U.</td>
<td>Sally</td>
<td>_x * 1.05</td>
</tr>
<tr>
<td></td>
<td>Sally</td>
<td>_x</td>
</tr>
</tbody>
</table>

Updates in QBE (3)

- “All employees who make less than 2000 receive a 5% raise”

<table>
<thead>
<tr>
<th>employee</th>
<th>name</th>
<th>salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>U.</td>
<td>_u</td>
<td>_x * 1.05</td>
</tr>
<tr>
<td></td>
<td>_u</td>
<td>_x</td>
</tr>
</tbody>
</table>

- Condition box: **_x < 2000**

- Note: QBE allows explicit specification of conditions using **condition boxes**