Set Theoretic Ops

- Duplicates (Frank Wong) eliminated.
- DEFAULT field names from first relation
Notes on Naming

- We unioned GOOD with WELL PAID.
- GOOD was first, so the attribute names from GOOD are used.
- Attributes can be renamed

```plaintext
GOOD_OR_WELL_PAID(FIRST, LNAME, PAY) ← GOOD ∪ WELL_PAID
```

Do Not Use the ρ Renaming Operator!!!

### Intersection

<table>
<thead>
<tr>
<th>GOOD</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FNAME LNAME</td>
<td>SAL</td>
</tr>
<tr>
<td>Frank Wong</td>
<td>40K</td>
</tr>
<tr>
<td>Alice Zelaya</td>
<td>25K</td>
</tr>
<tr>
<td>Joyce English</td>
<td>25K</td>
</tr>
<tr>
<td>Ahmad Jabbar</td>
<td>25K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WELL_PAID</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRST LAST</td>
<td>SAL</td>
</tr>
<tr>
<td>Frank Wong</td>
<td>40K</td>
</tr>
<tr>
<td>Jenny Wallace</td>
<td>43K</td>
</tr>
<tr>
<td>Ram Narayan</td>
<td>38K</td>
</tr>
<tr>
<td>James Borg</td>
<td>55K</td>
</tr>
</tbody>
</table>

```plaintext
GOOD_AND_WELL_PAID ← GOOD ∩ WELL_PAID
```

<table>
<thead>
<tr>
<th>GOOD_AND_WELL_PAID</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FNAME LNAME SAL</td>
<td></td>
</tr>
<tr>
<td>Frank Wong 40K</td>
<td></td>
</tr>
</tbody>
</table>
Set Difference

<table>
<thead>
<tr>
<th>GOOD</th>
<th>FNAME</th>
<th>LNAME</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frank</td>
<td>Wong</td>
<td>40K</td>
<td></td>
</tr>
<tr>
<td>Alice</td>
<td>Zelaya</td>
<td>25K</td>
<td></td>
</tr>
<tr>
<td>Joyce</td>
<td>English</td>
<td>25K</td>
<td></td>
</tr>
<tr>
<td>Ahmad</td>
<td>Jabbar</td>
<td>25K</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WELL_PAID</th>
<th>FIRST</th>
<th>LAST</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frank</td>
<td>Wong</td>
<td>40K</td>
<td></td>
</tr>
<tr>
<td>Jenny</td>
<td>Wallace</td>
<td>43K</td>
<td></td>
</tr>
<tr>
<td>Ram</td>
<td>Narayan</td>
<td>38K</td>
<td></td>
</tr>
<tr>
<td>James</td>
<td>Borg</td>
<td>55K</td>
<td></td>
</tr>
</tbody>
</table>

GOOD_NOT_WELL_PAID ← GOOD —— WELL_PAID

Everything in GOOD which is NOT ALSO in WELL_PAID

Notes on Difference

• Not Commutative: A - B ≠ B - A
• You can subtract things which are not there! ---
  \{A,B,C\} - \{C,D,E,F,G\} = \{A,B\}
Union Compatibility

- Operands to set ops must have the same domain
  - Fields have same domains
  - Fields in same order
  - Same number of fields
- Do not need to have same names.
- **WARNING:** due to a bug in RAO, set ops may fail unless the corresponding attributes of both operands have the same or similar names.

Why Union Compatibility?

Ordinary set ops do NOT require compatibility:

\[
\{“Mom”, , \} \cup \{ , , \} \text{ yields } \{“Mom”, , , \}
\]
Ordinary Set Op and Non-Compatible Operands

GOOD
FNAME LNAME SAL
Frank Wong 40K
Alice Zelaya 25K
Joyce English 25K
Ahmad Jabbar 25K

WELL_PAID
FIRST LAST SAL WGT
Frank Wong 40K 180
Jenny Wallace 43K 125
Ram Narayan 38K 173
James Borg 55K 265

GOOD  WELL_PAID
Frank Wong 40K
Alice Zelaya 25K
Joyce English 25K
Ahmad Jabbar 25K
Frank Wong 40K 180
Jenny Wallace 43K 125
Ram Narayan 38K 173
James Borg 55K 265

Legal Set but Not a Table Extent!

Why Not?

No Intension with Ordinary Set Op

One Domain (str,str,int)

Another Domain (str,str,int,int)

Database Set Op and Non-Compatible Operands

GOOD
FNAME LNAME SAL
Frank Wong 40K
Alice Zelaya 25K
Joyce English 25K
Ahmad Jabbar 25K

WELL_PAID
FIRST LAST SAL WGT
Frank Wong 40K 180
Jenny Wallace 43K 125
Ram Narayan 38K 173
James Borg 55K 265

GOOD  WELL_PAID
GOOD  WELL_PAID

GOOD OR WELL_PAID  GOOD  WELL_PAID

Syntax Error
(union compatibility)