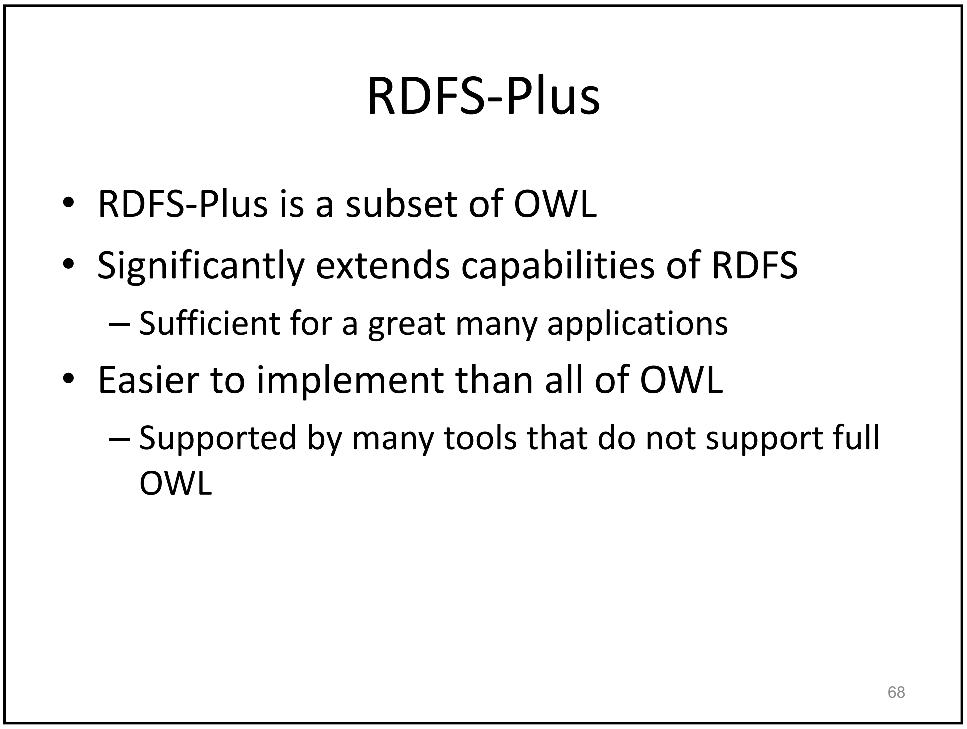
RDFS-PLUS

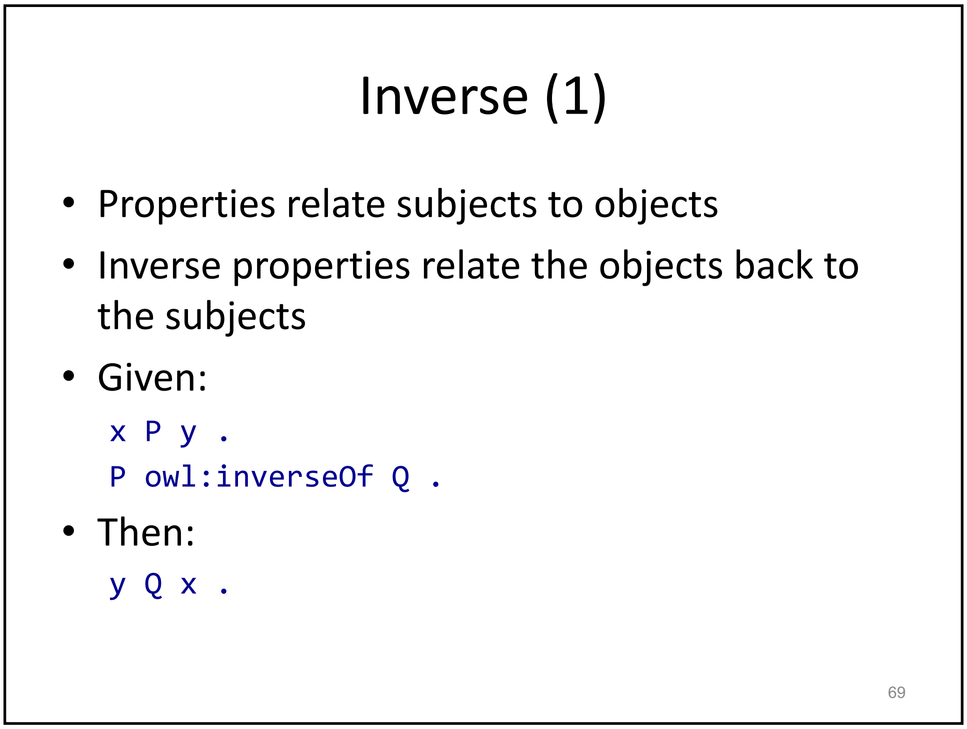
67



RDFS-Plus

- RDFS-Plus is a subset of OWL
- Significantly extends capabilities of RDFS
  - Sufficient for a great many applications
- Easier to implement than all of OWL
  - Supported by many tools that do not support full OWL

68



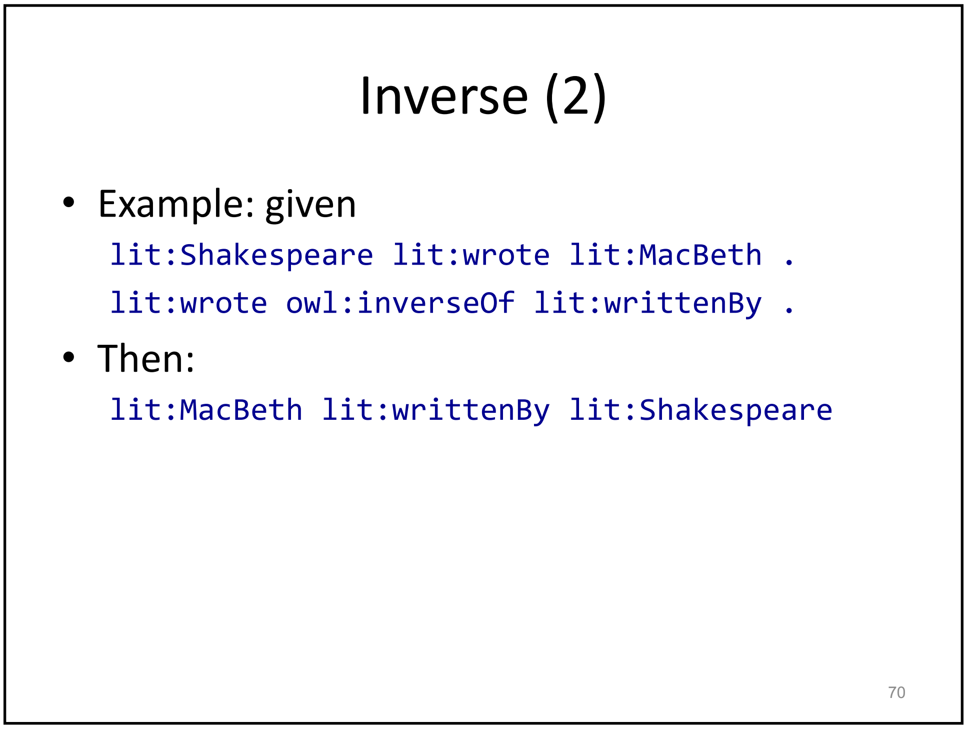
Inverse (1)

- Properties relate subjects to objects
- Inverse properties relate the objects back to the subjects
- Given:

```
x P y .
P owl:inverseOf Q .
```

- Then:

```
y Q x .
```



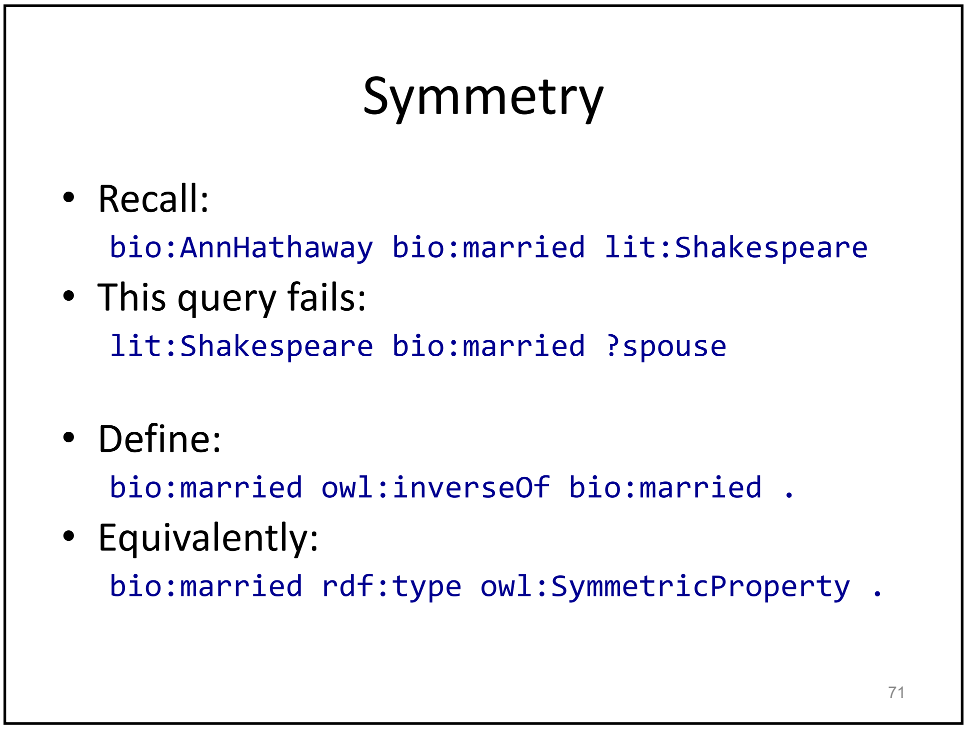
Inverse (2)

- Example: given

```
lit:Shakespeare lit:wrote lit:MacBeth .
lit:wrote owl:inverseOf lit:writtenBy .
```

- Then:

```
lit:MacBeth lit:writtenBy lit:Shakespeare
```



Symmetry

- Recall:

```
bio:AnnHathaway bio:married lit:Shakespeare
```

- This query fails:

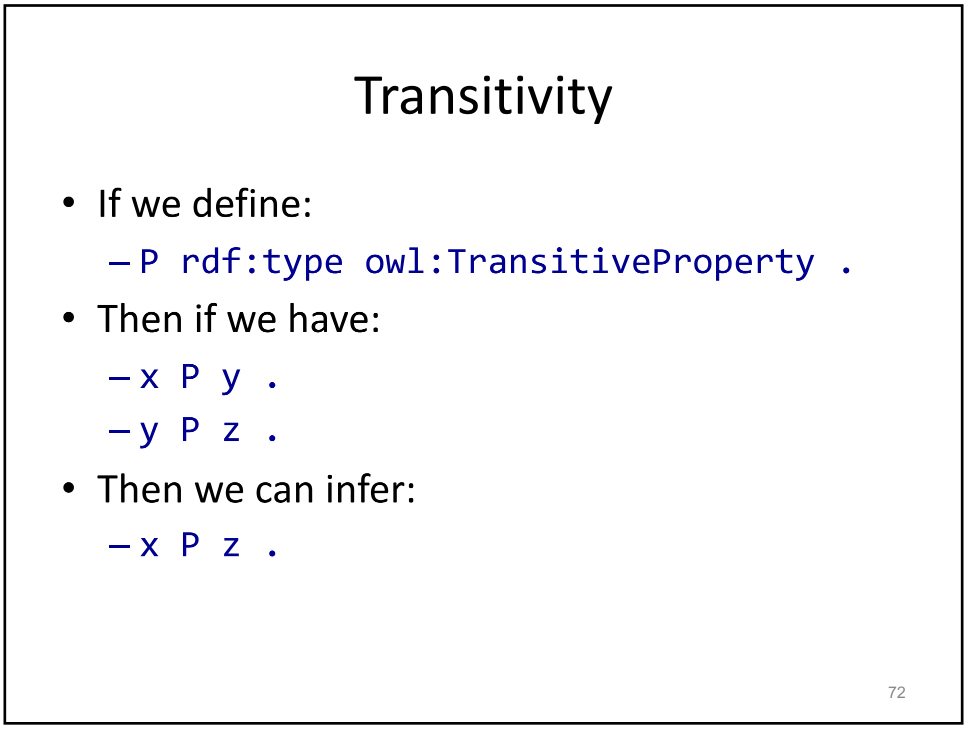
```
lit:Shakespeare bio:married ?spouse
```

- Define:

```
bio:married owl:inverseOf bio:married .
```

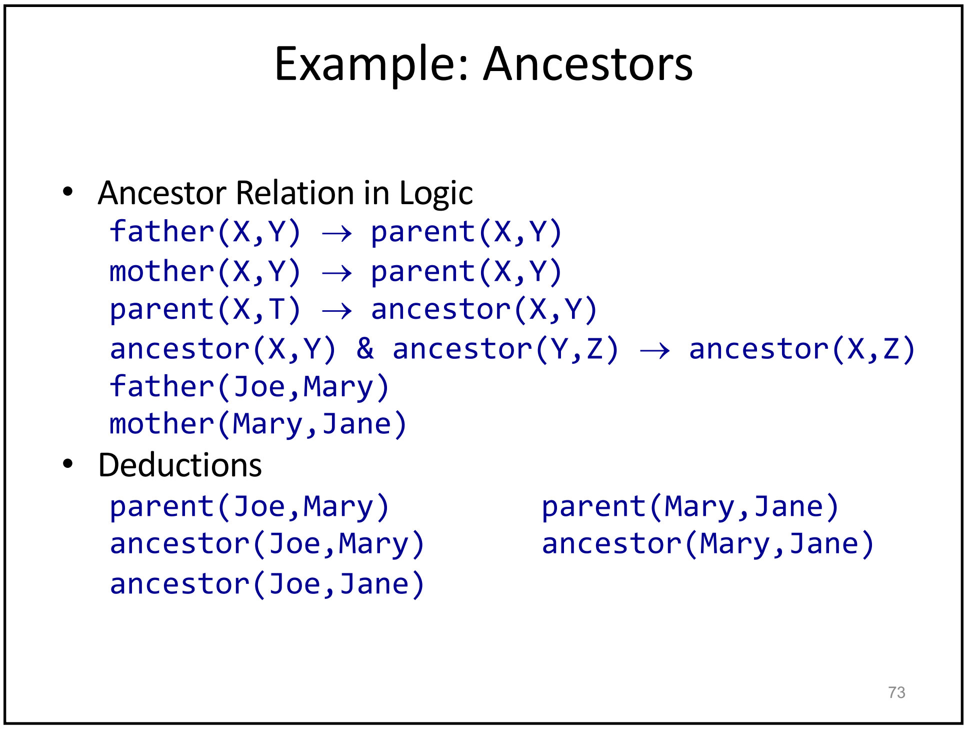
- Equivalently:

```
bio:married rdf:type owl:SymmetricProperty .
```



Transitivity

- If we define:
  - `P rdf:type owl:TransitiveProperty .`
- Then if we have:
  - `x P y .`
  - `y P z .`
- Then we can infer:
  - `x P z .`



Example: Ancestors

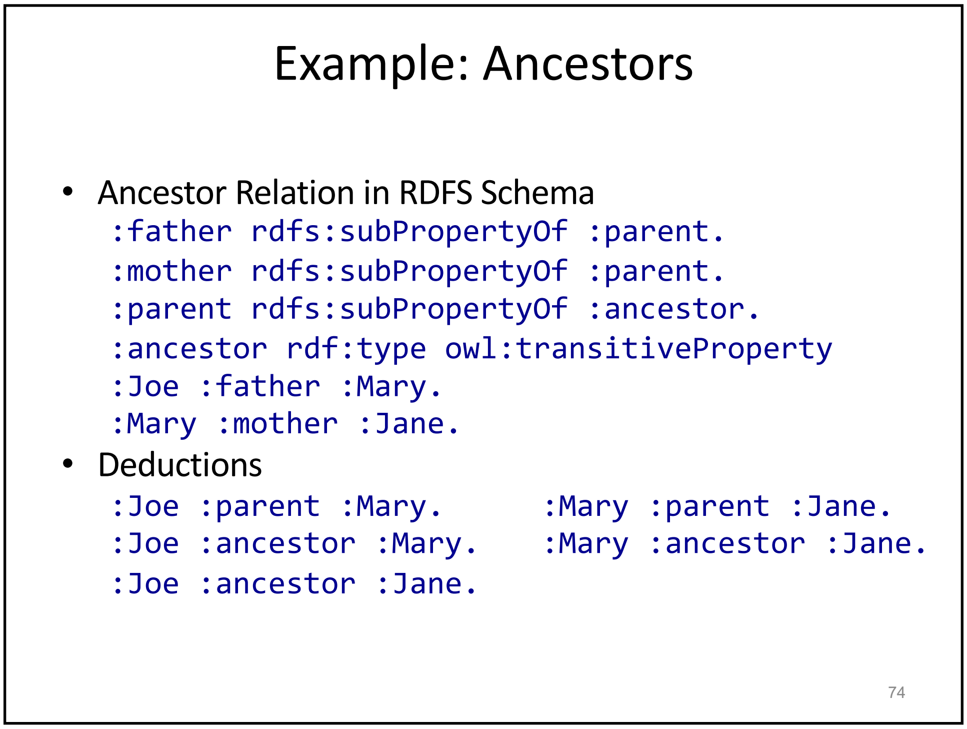
- Ancestor Relation in Logic

```
father(X,Y) → parent(X,Y)
mother(X,Y) → parent(X,Y)
parent(X,T) → ancestor(X,Y)
ancestor(X,Y) & ancestor(Y,Z) → ancestor(X,Z)
father(Joe,Mary)
mother(Mary,Jane)
```

- Deductions

```
parent(Joe,Mary)         parent(Mary,Jane)
ancestor(Joe,Mary)       ancestor(Mary,Jane)
ancestor(Joe,Jane)
```

73



Example: Ancestors

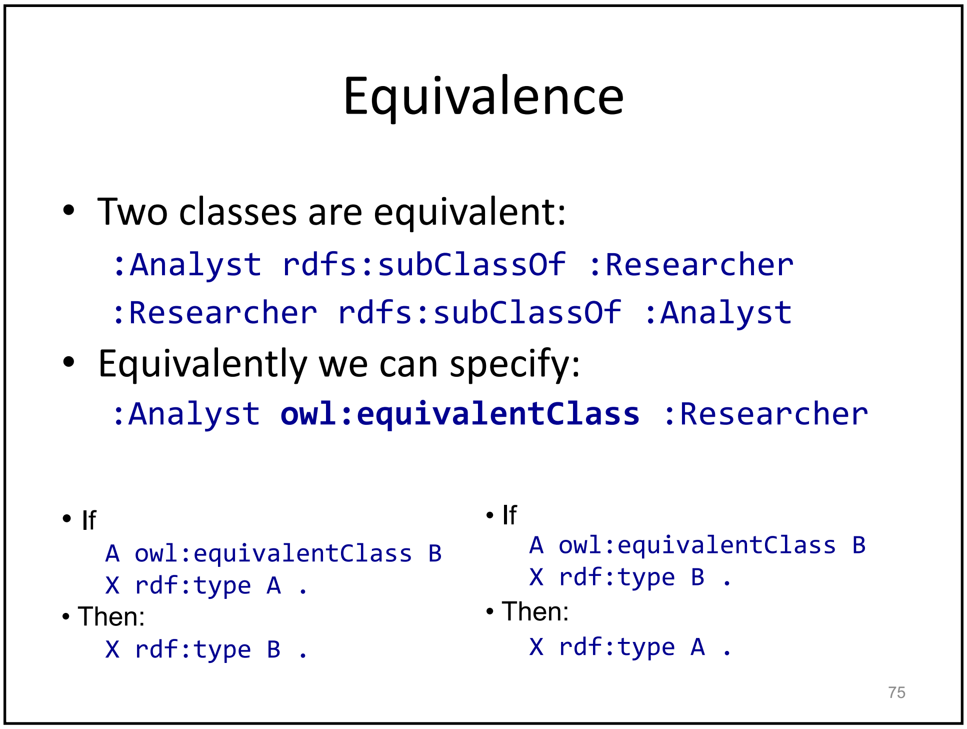
- Ancestor Relation in RDFS Schema

```
:father rdfs:subPropertyOf :parent.
:mother rdfs:subPropertyOf :parent.
:parent rdfs:subPropertyOf :ancestor.
:ancestor rdf:type owl:transitiveProperty
:Joe :father :Mary.
:Mary :mother :Jane.
```

- Deductions

```
:Joe :parent :Mary.      :Mary :parent :Jane.
:Joe :ancestor :Mary.    :Mary :ancestor :Jane.
:Joe :ancestor :Jane.
```

74



Equivalence

- Two classes are equivalent:

```
:Analyst rdfs:subClassOf :Researcher
:Researcher rdfs:subClassOf :Analyst
```

- Equivalently we can specify:

```
:Analyst owl:equivalentClass :Researcher
```

- If

```
A owl:equivalentClass B
X rdf:type A .
```

- Then:

```
X rdf:type B .
```

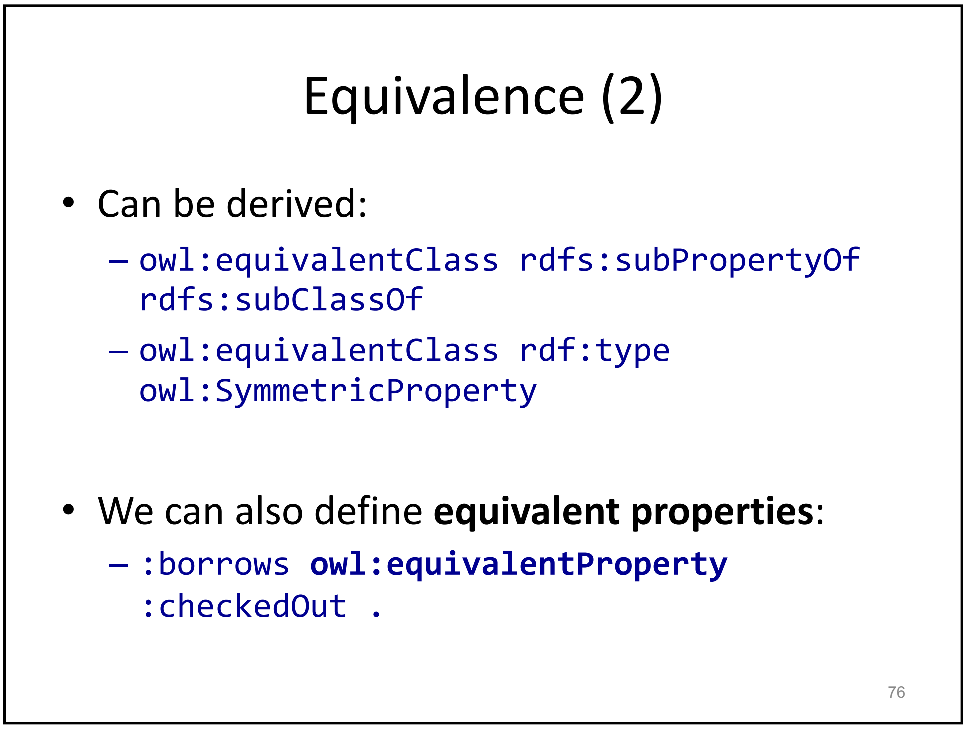
- If

```
A owl:equivalentClass B
X rdf:type B .
```

- Then:

```
X rdf:type A .
```

75



Equivalence (2)

- Can be derived:
  - `owl:equivalentClass rdfs:subPropertyOf rdfs:subClassOf`
  - `owl:equivalentClass rdf:type owl:SymmetricProperty`

- We can also define **equivalent properties**:
  - `:borrows owl:equivalentProperty :checkedOut .`

76

Equivalence (3)

- Equivalent individuals:

```
lit:Shakespeare lit:wrote lit:Hamlet .
spr:Hamnet spr:hasFather spr:WilliamShakspere .
```

- Did Hamnet’s father write “Hamlet”?

```
[spr:Hamnet spr:hasFather ?d .
?d lit:wrote lit:Hamlet . ]
```

- Yes if we identify:

```
spr:WilliamShakspere owl:sameAs
lit:Shakespeare .
```

Example: Reconciling Federated Databases

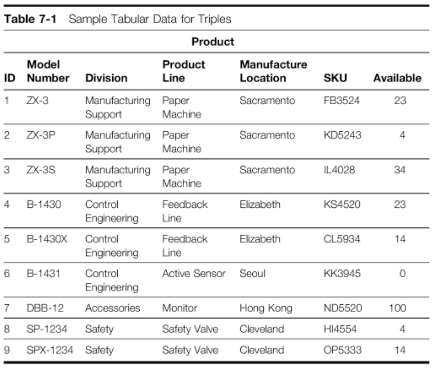


Table 7-1 Sample Tabular Data for Triples

			Product			
ID	Model Number	Division	Product Line	Manufacture Location	SKU	Available
1	ZX-3	Manufacturing Support	Paper Machine	Sacramento	FB3524	23
2	ZX-3P	Manufacturing Support	Paper Machine	Sacramento	KD5243	4
3	ZX-3S	Manufacturing Support	Paper Machine	Sacramento	IL4028	34
4	B-1430	Control Engineering	Feedback Line	Elizabeth	KS4520	23
5	B-1430X	Control Engineering	Feedback Line	Elizabeth	CL5934	14
6	B-1431	Control Engineering	Active Sensor	Seoul	KK3945	0
7	DBB-12	Accessories	Monitor	Hong Kong	ND5520	100
8	SP-1234	Safety	Safety Valve	Cleveland	HI4554	4
9	SPX-1234	Safety	Safety Valve	Cleveland	OP5333	14

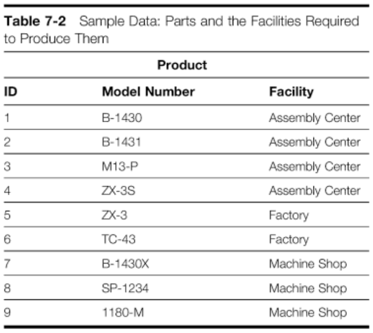
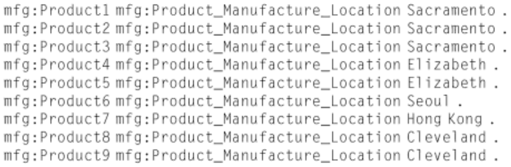


Table 7-2 Sample Data: Parts and the Facilities Required to Produce Them

	Product	
ID	Model Number	Facility
1	B-1430	Assembly Center
2	B-1431	Assembly Center
3	M13-P	Assembly Center
4	ZX-3S	Assembly Center
5	ZX-3	Factory
6	TC-43	Factory
7	B-1430X	Machine Shop
8	SP-1234	Machine Shop
9	1180-M	Machine Shop

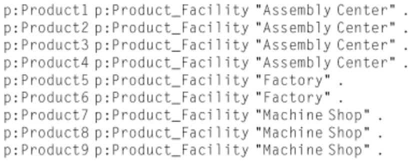
Two different databases with product information

Example: Reconciling Federated Databases (2)



```
mfg:Product1 mfg:Product_Manufacture_Location Sacramento .
mfg:Product2 mfg:Product_Manufacture_Location Sacramento .
mfg:Product3 mfg:Product_Manufacture_Location Sacramento .
mfg:Product4 mfg:Product_Manufacture_Location Elizabeth .
mfg:Product5 mfg:Product_Manufacture_Location Elizabeth .
mfg:Product6 mfg:Product_Manufacture_Location Seoul .
mfg:Product7 mfg:Product_Manufacture_Location Hong Kong .
mfg:Product8 mfg:Product_Manufacture_Location Cleveland .
mfg:Product9 mfg:Product_Manufacture_Location Cleveland .
```

Triples from first database



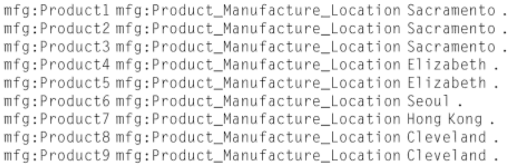
```
p:Product1 p:Product_Facility "Assembly Center" .
p:Product2 p:Product_Facility "Assembly Center" .
p:Product3 p:Product_Facility "Assembly Center" .
p:Product4 p:Product_Facility "Assembly Center" .
p:Product5 p:Product_Facility "Factory" .
p:Product6 p:Product_Facility "Factory" .
p:Product7 p:Product_Facility "Machine Shop" .
p:Product8 p:Product_Facility "Machine Shop" .
p:Product9 p:Product_Facility "Machine Shop" .
```

Triples from second database

79

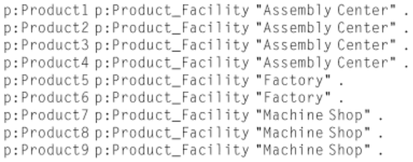
79

Example: Reconciling Federated Databases (2)



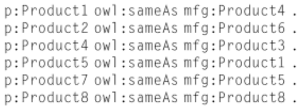
```
mfg:Product1 mfg:Product_Manufacture_Location Sacramento .
mfg:Product2 mfg:Product_Manufacture_Location Sacramento .
mfg:Product3 mfg:Product_Manufacture_Location Sacramento .
mfg:Product4 mfg:Product_Manufacture_Location Elizabeth .
mfg:Product5 mfg:Product_Manufacture_Location Elizabeth .
mfg:Product6 mfg:Product_Manufacture_Location Seoul .
mfg:Product7 mfg:Product_Manufacture_Location Hong Kong .
mfg:Product8 mfg:Product_Manufacture_Location Cleveland .
mfg:Product9 mfg:Product_Manufacture_Location Cleveland .
```

Triples from first database



```
p:Product1 p:Product_Facility "Assembly Center" .
p:Product2 p:Product_Facility "Assembly Center" .
p:Product3 p:Product_Facility "Assembly Center" .
p:Product4 p:Product_Facility "Assembly Center" .
p:Product5 p:Product_Facility "Factory" .
p:Product6 p:Product_Facility "Factory" .
p:Product7 p:Product_Facility "Machine Shop" .
p:Product8 p:Product_Facility "Machine Shop" .
p:Product9 p:Product_Facility "Machine Shop" .
```

Triples from second database

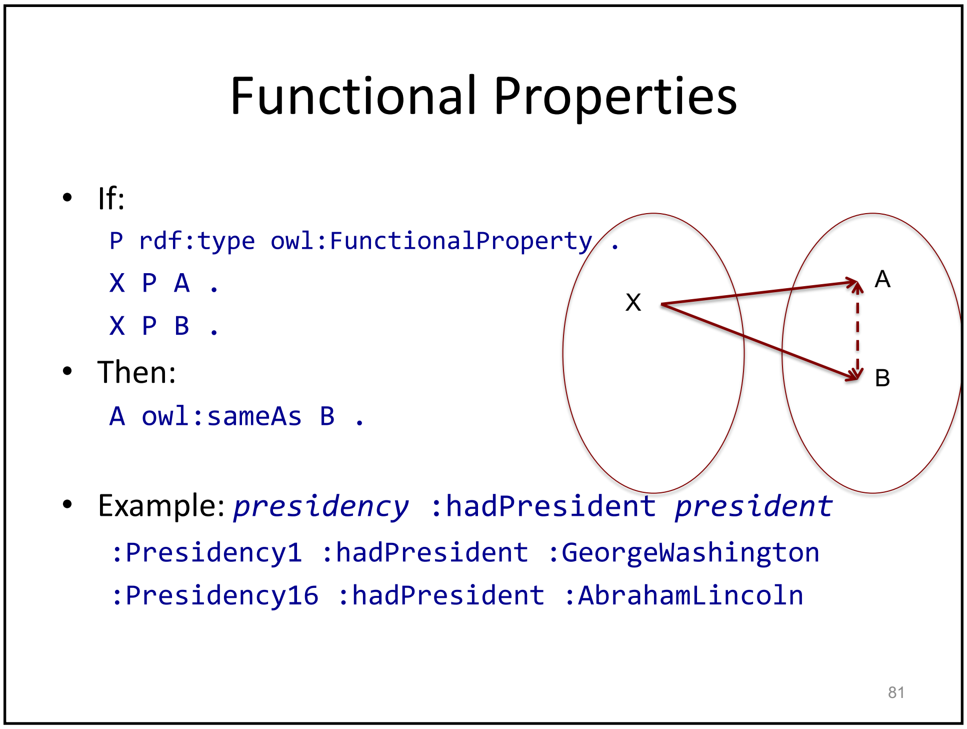


```
p:Product1 owl:sameAs mfg:Product4 .
p:Product2 owl:sameAs mfg:Product6 .
p:Product4 owl:sameAs mfg:Product3 .
p:Product5 owl:sameAs mfg:Product1 .
p:Product7 owl:sameAs mfg:Product5 .
p:Product8 owl:sameAs mfg:Product8 .
```

Identifying rows

80

80



Functional Properties

- If:

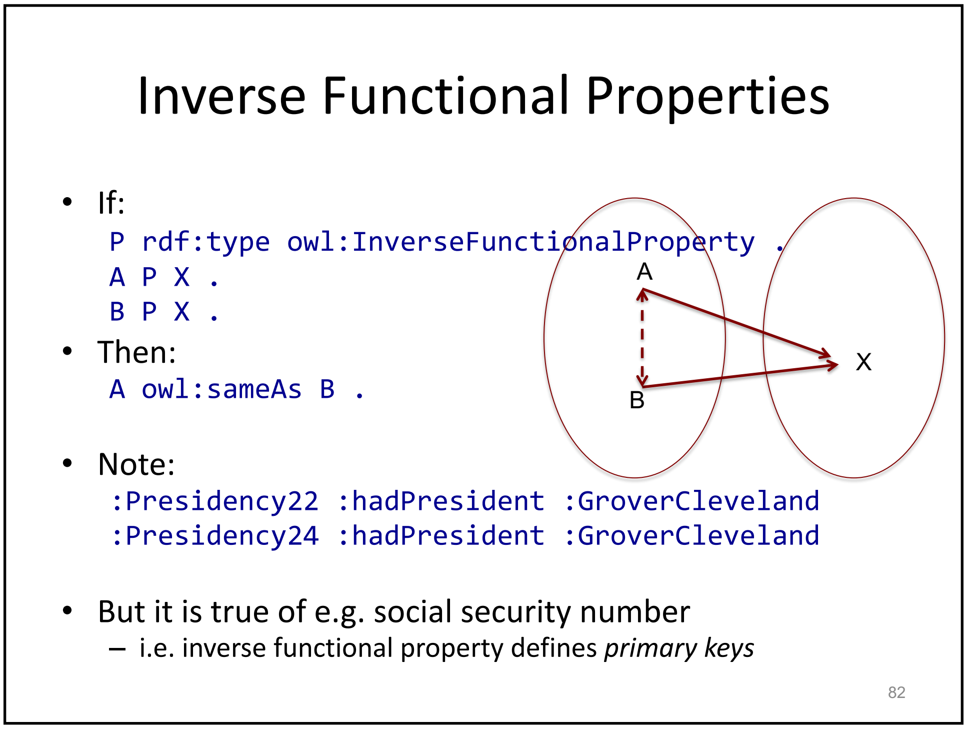
```
P rdf:type owl:FunctionalProperty .
X P A .
X P B .
```

- Then:

```
A owl:sameAs B .
```

- Example: *presidency* :hadPresident *president*

```
:Presidency1 :hadPresident :GeorgeWashington
:Presidency16 :hadPresident :AbrahamLincoln
```



Inverse Functional Properties

- If:

```
P rdf:type owl:InverseFunctionalProperty .
A P X .
B P X .
```

- Then:

```
A owl:sameAs B .
```

- Note:

```
:Presidency22 :hadPresident :GroverCleveland
:Presidency24 :hadPresident :GroverCleveland
```

- But it is true of e.g. social security number
  - i.e. inverse functional property defines *primary keys*

Example

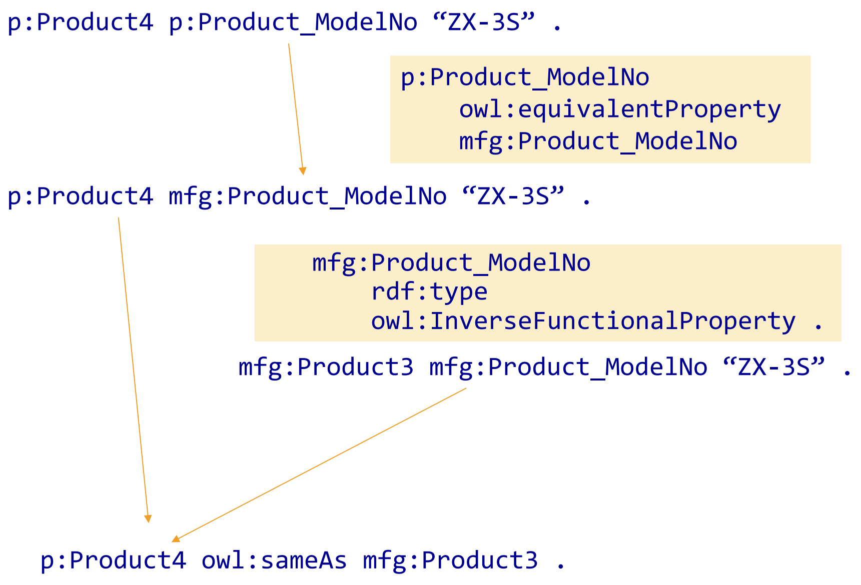
- Use inverse functional properties to automatically reconcile federated databases

- Product model number is a primary key:

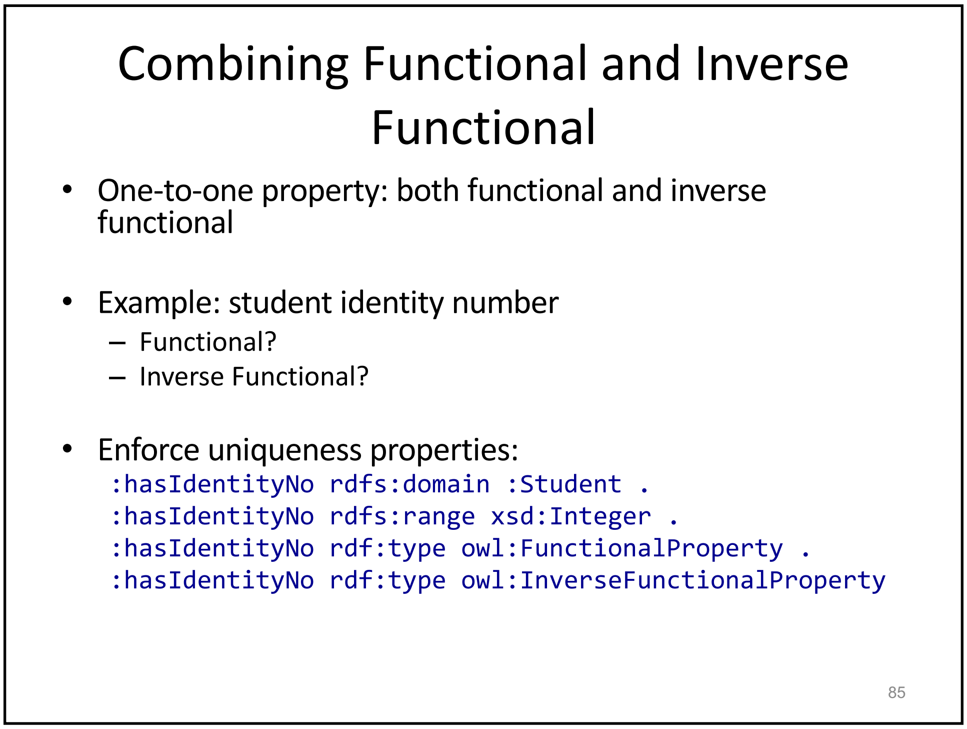
```
mfg:Product_ModelNo
    rdf:type owl:InverseFunctionalProperty .
```

- Equate model numbers:

```
p:Product_ModelNo
    owl:equivalentProperty mfg:Product_ModelNo
.
```



```
p:Product4 p:Product_ModelNo “ZX-3S” .

p:Product_ModelNo
    owl:equivalentProperty
    mfg:Product_ModelNo

p:Product4 mfg:Product_ModelNo “ZX-3S” .

mfg:Product_ModelNo
    rdf:type
    owl:InverseFunctionalProperty .

mfg:Product3 mfg:Product_ModelNo “ZX-3S” .

p:Product4 owl:sameAs mfg:Product3 .
```



Combining Functional and Inverse Functional

- One-to-one property: both functional and inverse functional

- Example: student identity number
  - Functional?
  - Inverse Functional?

- Enforce uniqueness properties:

```
:hasIdentityNo rdfs:domain :Student .
:hasIdentityNo rdfs:range xsd:Integer .
:hasIdentityNo rdf:type owl:FunctionalProperty .
:hasIdentityNo rdf:type owl:InverseFunctionalProperty
```

85