

Copyright ©1998 Timothy Howard Merrett

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation in a prominent place. Copyright for components of this work owned by others than T. H. Merrett must be honoured. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or fee. Request permission to republish from: T. H. Merrett, School of Computer Science, McGill University, fax 514 398 3883.

The author gratefully acknowledges support from the taxpayers of Québec and of Canada who have paid his salary and research grants while this work was developed at McGill University, and from his students (who built the implementations and investigated the data structures and algorithms) and their funding agencies.

T. H. Merrett

©98/10

Updates

Additions

<i>Class</i> (Item	<i>Type)</i>	<i>ReClass</i> (Item	<i>Type)</i>
Yarn	A	Yarn	A
String	A	String	B
Ball	B	Top	A
Sandal	C		

Class <- *Class* **ujoin** where *Item*="Top" in *ReClass*;

Class <+ **where** *Item*="Top" in *ReClass*;

update *Class* **add** where *Item*="Top" in *ReClass*;

<i>Class</i> (Item	<i>Type)</i>
Yarn	A
String	A
Ball	B
Sandal	C
Top	A

Deletions

<i>Class</i>	<i>(Item</i>	<i>Type)</i>	<i>ReClass</i>	<i>(Item</i>	<i>Type)</i>
	Yarn	A		Yarn	A
	String	A		String	B
	Ball	B		Top	A
	Sandal	C			

Class ← *Class* **djoin** *ReClass*;

update *Class* **delete** *ReClass*;

<i>Class</i>	<i>(Item</i>	<i>Type)</i>
	String	A
	Ball	B
	Sandal	C

Changes

<i>Class</i> (<i>Item</i>	<i>Type</i>)	<i>ReClass</i> (<i>Item</i>	<i>Type</i>)
Yarn	A	Yarn	A
String	A	String	B
Ball	B	Top	A
Sandal	C		

update *Class* change *Type* ← "B" using **ijoin** on *ReClass*;

<i>Class</i> ijoin	<i>ReClass</i> (<i>Item</i>	<i>Type</i>)
	Yarn	A

<i>Class</i> (<i>Item</i>	<i>Type</i>)
Yarn	B
String	A
Ball	B
Sandal	C

update *Class* change *Type* ← "B" using *ReClass*;

<i>Class</i> (<i>Item</i> <i>Type</i>)	<i>ReClass</i> (<i>Item</i> <i>Type</i>)
Yarn A	Yarn A
String A	String B
Ball B	Top A
Sandal C	

update Class change *Type* ← "B" **using** [*Item*] **in** *ReClass*;

<i>Class</i> ijoin [<i>Item</i>] in <i>ReClass</i> (<i>Item</i> <i>Type</i>)
Yarn A
String A

<i>Class</i> (<i>Item</i> <i>Type</i>)
Yarn B
String B
Ball B
Sandal C

<i>Class</i>	<i>(Item</i>	<i>Type)</i>	<i>ReClass</i>	<i>(Item</i>	<i>Type)</i>
	Yarn	A		Yarn	A
	String	A		String	B
	Ball	B		Top	A
	Sandal	C			

let *NewType* **be** *Type*;
update *Class* **change** *Type* ← *NewType* **using**
 [*Item*, *NewType*] **in** *ReClass*;

<i>Class</i>	ijoin	[<i>Item</i> ,	<i>NewType]</i>	in	<i>ReClass</i>
		<i>(Item</i>	<i>Type</i>	<i>NewType)</i>	
	Yarn	A		A	
	String	A		B	

<i>Class</i>	<i>(Item</i>	<i>Type)</i>
	Yarn	A
	String	B
	Ball	B
	Sandal	C

<i>Class</i>	<i>(Item</i>	<i>Type)</i>	<i>ReClass</i>	<i>(Item</i>	<i>Type)</i>
	Yarn	A		Yarn	A
	String	A		String	B
	Ball	B		Top	A
	Sandal	C			

let *NewType* **be** *Type*;
update *Class* **change** *Type* ← *NewType* **using** **ujoin**
on [*Item*, *NewType*] **in** *ReClass*;

<i>Class</i>	ujoin	<i>[Item,</i>	<i>NewType]</i>	in	<i>ReClass</i>
		<i>(Item</i>	<i>Type</i>	<i>NewType)</i>	
	Yarn	A		A	
	String	A		B	
	Ball	B			
	Sandal	C			
	Top			A	

<i>Class</i>	<i>(Item</i>	<i>Type)</i>
	Yarn	A
	String	B
	Ball	B
	Sandal	C
	Top	A

<i>Class</i>	<i>(Item</i>	<i>Type)</i>	<i>ReClass</i>	<i>(Item</i>	<i>Type)</i>
Yarn		A	Yarn		A
String		A	String		B
Ball		B	Top		A
Sandal		C			

update *Class* change *Type* ← "B" using **djoin** on *ReClass*;

<i>Class</i>	djoin	<i>ReClass</i>	<i>(Item</i>	<i>Type)</i>
		String		A
		Ball		B
		Sandal		C

<i>Class</i>	<i>(Item</i>	<i>Type)</i>
Yarn		A
String		B
Ball		B
Sandal		B

<i>Class</i>	<i>(Item</i>	<i>Type)</i>
Yarn		A
String		A
Ball		B
Sandal		C

update Class change *Type* ← "B";

<i>Class</i>	<i>(Item</i>	<i>Type)</i>
Yarn		B
String		B
Ball		B
Sandal		B

let *NewType* **be** **if** *Type*="C" **then** "B" **else** *Type*;
update Class change *Type* ← *NewType*;

<i>Class</i>	<i>(Item</i>	<i>Type)</i>
Yarn		A
String		A
Ball		B
Sandal		B

let *NewType* **be** **if** *Item*="Sandal" **then** "B" **else** *Type*;
update Class change *Type* ← *NewType*;

<i>Class</i>	<i>(Item</i>	<i>Type)</i>
	Yarn	A
	String	A
	Ball	B
	Sandal	C

update *Class* change *Type* ← "A" using
where *Comp*="Playsew" **in** *Supply*;

Class *ijoin* **where** *Comp*="Playsew" **in** *Supply*
(*Item* *Type* *Comp* *Dept* *Vol*)

Yarn	A	Playsew	Rug	17
Yarn	A	Playsew	Toy	20
String	A	Playsew	Toy	10
String	A	Playsew	Shoe	5
Ball	B	Playsew	Toy	2

<i>Class</i>	<i>(Item</i>	<i>Type)</i>
	Yarn	A
	String	A
	Ball	A
	Sandal	C

<i>Class</i>	<i>(Item</i>	<i>Type)</i>
Yarn	A	
String	A	
Ball	B	
Sandal	C	

update *Class change Type* ← "B" **using**
Supply ijoin where Floor=2 in Loc;

<i>Class ijoin Supply ijoin where Floor=2 in Loc</i>					
<i>(Item</i>	<i>Type</i>	<i>Comp</i>	<i>Dept</i>	<i>Vol</i>	<i>Floor)</i>
Yarn	A	Domtex	Rug	10	2
Yarn	A	Playsew	Rug	17	2
String	A	Domtex	Rug	5	2
String	A	Playsew	Shoe	5	2
String	A	Shoeco	Shoe	15	2

<i>Class</i>	<i>(Item</i>	<i>Type)</i>
Yarn	B	
String	B	
Ball	B	
Sandal	C	

<i>Class</i>	<i>Item</i>	<i>Type</i>
	Yarn	A
	String	A
	Ball	B
	Sandal	C

update Class change *Type* ← "B" **using** [*Item*]
where {(# ≥ 2) *Comp*, (# > 1) *Dept*} **in** *Supply*;

<i>Class</i>	<i>ijoin</i>	{("String")}	<i>Item</i>	<i>Type</i>
			String	A

<i>Class</i>	<i>Item</i>	<i>Type</i>
	Yarn	A
	String	B
	Ball	B
	Sandal	C