### ORACLE

### **Multi-Edge support and Future-Proofing**

#### Souripriya Das, Ph.D.

Architect, Spatial and Graph, Oracle America Inc.

January 19, 2023



### Background: Edges as Vertices, Multigraph, Multi-Edge (or Parallel Edges)

• A Line Graph (LG) converts edges to vertices and then eliminates the original vertices and ...



RDF-star, keeps, and allows unrestricted use of, <u>both edge-vertices and the original vertices</u>, as vertices.
 Property Graph (PG) supports it too, but <u>limits edge-vertices to only connect to scalar values</u>.

### **Labeled Multidigraphs are Not Uncommon in Practice**

Examples:

- :servedAs :POTUS" (Cleveland Grover did two non-consecutive terms)
- :deposit :myBankAccount (multiple transactions by same person to same account)
- :called :mySister (call data records: multiple calls by same person to his/her sister)
- :hasManager :myManager (multiple stints)
- :won :Wimbledon (same person wins multiple times)
- :won :SoccerWorldCup (same country wins multiple times)
- etc.

### Modeling Multi-Edge and Handling Transition of a Property to Multi-Edge

Da	ta		Column Names						<u>R[</u>	<u>)F-st</u>	<u>ar</u>	Pro	pei	rty Ta	ables	:0	ccurre	enceOf	as	neede	ed
X	y color	type	1	1. name $\rightarrow$ implicit or explicit					MS	iid	x	у	Qf	oid	iid	lor	oid	color	/pe	oid	dtype
ΑI	B red		2	2. iid $\rightarrow$ implicit id					u o	akb	А	В	JCe			0	akb	red	dt	akb	
B	C blue		3	oid		irre	ence <mark>id</mark>		<del>*</del> .	bkc	В	С	rrei				bkc	blue		bkc	
ΒI	D blue		4	I. x →	subjec	t o	f triple	2		bkd	В	D	cul				bkd	blue		bkd	
CI	D green		5	. y →	object	of	triple			ckd	С	D	00	cd1	ckd		cd1	green		cd1	
C	<b>D</b> blue													cd2	ckd		cd2	blue		cd2	
<u>R</u>	<u>DFn</u> Pro	operty	Tables: <b>no</b> :occurrenceOf					D		) Jr	Dro	201	rty T-	hloc		COLIKK					
				0.001	•					<u> </u>		PIO	per		Dies	:0	ccurre	enceOr	AL	WAYS	)
WS	name	x y	lor	name	color	/be	name	dtype	SM	iid		y	bei fo	oid	iid		oid	color	AL Jod/	oid	dtype
KNOWS	<mark>name</mark> akb	xy AB	:color	name akb	<b>color</b> red	dtype	name akb	dtype 	Swoux	iid akb	x A	y B	nceOf	oid ab1	iid akb	:color	oid ab1	color red	dtype	oid ab1	dtype 
:knows	name akb bkc	x y A B B C	:color	name akb bkc	color red blue	:dtype	name akb bkc	<b>dtype</b> 	:knows	iid akb bkc	X A B	y B C	rrenceOf	oid ab1 bc1	iid akb bkc	:color	oid ab1 bc1	color red blue	:dtype	oid ab1 bc1	dtype 
:knows	name akb bkc bkd	x y A B B C B D	:color	name akb bkc bkd	color red blue blue	:dtype	name akb bkc bkd	<b>dtype</b> 	:knows	iid akb bkc bkd	X A B B	y B C D	courrenceOf	oid ab1 bc1 bd1	iid akb bkc bkd	:color	oid ab1 bc1 bd1	color red blue blue	:dtype	oid ab1 bc1 bd1	dtype 
:knows	name akb bkc bkd ckd	x         y           A         B           B         C           B         D           C         D	:color	name akb bkc bkd ckd	color red blue blue green	:dtype	name akb bkc bkd ckd	<b>dtype</b>	:knows	iid akb bkc bkd ckd	A B C	y B C D	:occurrenceOf	oid ab1 bc1 bd1 cd1	iid akb bkc bkd ckd	:color	oid ab1 bc1 bd1 cd1	color red blue blue green	:dtype	oid ab1 bc1 bd1 cd1	<b>dtype</b>

## RDFn Named Triples:

### Data Size, Burden on Creator, Query Complexity/Efficiency, Future-Proof?

D	ata	а		Key Measures
x	У	color	type	1. # of core triples: $4 + 1 = 5$
А	В	red		2. # of custom IRIs: 1
В	С	blue		3. Qry Complexity: Simple.
В	D	blue		4. # of triple-patterns: 1
С	D	green		5. Is future-proof? YES
C	D	blue		

<u>R</u> [	<u>DFn</u> Pro	op	ert	у Та	bles: <b>n</b>	<b>o</b> :occu	irre	enceOf						
WS	name	X	у	lor	name	color	,pe	name	dtype					
(NO	akb	А	В	00	akb	red	dty	akb						
<b>—</b>	bkc	В	С		bkc	blue		bkc						
	bkd	В	D		bkd	blue		bkd						
	ckd C D ckd green ckd													
	cd2         C         D         cd2         blue         cd2													

:A :knows :B . :B :knows :C . :B :knows :D . :C :knows :D . :C :knows :D   :cd2 .	Data: Core triples only

Query: Count occurrences of (asserted) :knows edges.

```
SELECT (count(*) as ?cnt) {
  ?x :knows :?y
```

E	BEFORE multi-edge
---	-------------------

AFTER multi-edge

... same as BEFORE ...

### RDF-star with :occurrenceOf <u>for extra edges</u> in multi-edges: Data Size, Burden on Creator, Query Complexity/Efficiency, Future-Proof?

Data	Key Measures		RDF-st	<u>ar</u> Pro	perty	Tables	5::0	ccurr	enceOf	as	neede	ed		
x y color type	1. # of core triples: $4 + 1 = 5$		S iid	хy	joid	iid	<u> </u>	oid	color	/pe	oid	dtype		
A B red	2. # of custom IRIs: 1		O akb	AB	UCE			akb	red	dty	akb			
B C blue	3. Qry Complexity: UNION.			3. Qry Complexity: UNION.	<b>b</b> kc	ВC	rrei			<u>bkc</u>	blue		bkc	
B D blue	4. # of triple-patterns: 2.		bkd	B D	CCU			bkd	blue		bkd	_		
C D green	5. Is future-proof? NO		ckd	C D	0.			ckd	green		ckd			
C D blue					cd2	ckd		cd2	blue		cd2			
:A :knows :B . :B :knows :C . :B :knows :D . :C :knows :D .	Data: Core triples only	Qu SEl ?x	ery: Cou LECT (co c :knows	int oc ount(* :?y	curren ) as ?c	<mark>ces of</mark> nt) {	(as	serte	d) :knov	ws	edges			
.cd2 .occurrence	of << ·C ·knows ·D >>	}						BE	FORE n	nult	ti-edg	e		
6 Copyright © 2023 0	racle and/or its affiliates.	SEI { î	LECT (co ?x :know NION { ?	ount(* /s :?y } /occ :c	) as ?c	nt) { nceOf	f <<	A < ?x :k	FTER m	nult y >:	ti-edg >}	e		

### RDF-star with :occurrenceOf <u>for all edges</u> in multi-edges: Data Size, Burden on Creator, Query Complexity/Efficiency, Future-Proof?

Data	Key Measures	RDF-st	ar P	rope	erty Ta	ables	:000	urre	nceOf	as	need	ed
x y color type	1. # of core triples: 4 + 2*1 = 6	S iid	x y	J O	oid	iid	<u>o</u>	id	color	/pe	oid	dtype
A B red	2. # of custom IRIs: 2*1 = 2	O akb	AE				O a	kb	red	dty	akb	
B C blue	3. Qry Complexity: <b>OPTIONAL</b> .	bkc	BC	rrel			b	kc	blue		bkc	
B D blue	4. # of triple-patterns: 2.	bkd	BC				b	kd	blue		bkd	
C D green	5. Is future-proof? NO	ckd	CD		cd1	ckd	C	d1	green		cd1	_
C D blue					cd2	ckd	C	d2	blue		cd2	
:A :knows :B . :B :knows :C . :B :knows :D . :C :knows :D .	Data: Core triples only SI	uery: Cou ELECT (ce ?x :knows	unt o ount ; :?y	ссиі (*) а	rrence is ?cn	es of t) {	(asse	erted BEF	l) :knov ORE m	vs ( nult	edges :i-edg	5. ge
:cd1:occurrence :cd2:occurrence	2Of << :C :knows :D >> .			(*) -	- 2	4) (		AF	TER m	nult	i-edg	je
7 Copyright © 2023 O	racle and/or its affiliates.	Price (C)	5unt ; :?y L { ?	(*) a <mark>occ</mark>	is ?cn :occu	t) { rrenc	eOf ·	<< ?>	x :knov	VS î	Py >>]	0

### RDF-star with :occurrenceOf ALWAYS (for all edges): Data Size, Burden on Creator, Query Complexity/Efficiency, Future-Proof?

D	ata	a		Key Measures
X	У	color	type	1. # of core triples: 4*2 + 1 = 9
А	В	red		2. # of custom IRIs: 4 + 1 = 5
В	С	blue		3. Qry Complexity: Simple.
В	D	blue		4. # of triple-patterns: 2.
С	D	green		5. Is future-proof? YES
С	D	blue		

Query: Count occurrences of (asserted) :knows edges. SELECT (count(\*) as ?cnt) { ?x :knows :?y . ?occ :occurrenceOf << ?x :knows ?y >> } BEFORE multi-edge

AFTER multi-edge

... same as BEFORE ...

Data: Core triples only :A :knows :B . :ab1 :occurrenceOf << :A :knows :B >> . :B :knows :C . :bc1 :occurrenceOf << :B :knows :C >> . :B :knows :D . :bd1 :occurrenceOf << :B :knows :D >> . :C :knows :D . :cd1 :occurrenceOf << :C :knows :D >> .

:cd2 :occurrenceOf << :C :knows :D >> .



# **Comparison of Key Measures:** RDFn Named Triples vs. RDF-star Alternatives Data Size, Burden on Creator, Query Complexity/Efficiency, Future-Proof?

		riples	ceOf, edges edges	ceOf,		ceOf, <u>ses</u>	<u>R[</u>	<u>DF-st</u>	ar	Pro	pe	rty Ta	ables:	:0	ccurre	enceOf	as	neede	ed
		] in ned T	urren extra ( iulti-e	urren II ede	ti-edg	urren all edg	SWC	iid	x	У	eOf	oid	iid	olor	oid	color	ype	oid	dtype
Key	Measures	RDF Nan	in m	iocc for a		<u>for a</u>	kno	akb	Α	В	enco			0	akb	red	:dt	akb	
1.	# of core triples	5 = 4+	I <b>5</b> = 4+1	6 = 4	+ 2*1	<b>9</b> = 2*4 + 1		bkc	В	С	urre				bkc	blue		bkc	
2.	# of custom IRIs	1	1	2		5		bkd	В	D					bkd	blue		bkd	
3.	Query Complexity	Simpl		0P11 2	IONAL	Simple		ckd	С	D	ŏ	cd1	ckd		cd1	green		cd1	
<b>4</b> . 5.	ls future-proof?	YES	NO	NO		YES						cd2	ckd	) [	cd2	blue		cd2	
Ē	<u>RDFn</u> Property	Tables:	<b>no</b> :occ	urre	nceOf	F	<u>R[</u>	DF-st	ar	Pro	pe	rty Ta	ables:	:0	ccurre	enceOf	AL	WAYS	5
	<u>RDFn</u> Property name x y	Tables: ຼັວ <mark>ຼົ nam</mark>	no :occ e color	currei	nceOf name	dtype	RI SM	DF-st iid	ar x	Pro y	pe JO	rty Ta <mark>oid</mark>	ables: iid	lor .	ccurre <mark>oid</mark>	enceOf color	AL be	WAYS oid	dtype
	RDFn Property name x y akb A B	Tables:	no :occ e color red	urre dtype	nceOf name akb	dtype	RI SMOUX	DF-st iid akb	ar x A	Pro y B	nceOf a	rty Ta oid ab1	ables: iid akb	:color o:	ccurre oid ab1	enceOf color red	dtype P	WAYS oid ab1	dtype 
	RDFnPropertynamexyakbABbkcBC	Tables:	no :occ e color red blue	urrei dtype	nceOf name akb bkc	dtype	RI swouy:	DF-st iid akb bkc	ar × A B	Pro y B C	rrenceOf a	rty Ta oid ab1 bc1	ables: iid akb bkc	:color	oid ab1 bc1	color red blue	:dtype	WAYS oid ab1 bc1	dtype 
F smort	Non-StructureNon-StructurenamexyakbABbkcBCbkdBD	Tables: Definition Definitio	no :occ e color red blue blue	urrei dtype	nceOf name akb bkc bkd	 	RI smount:	DF-st iid akb bkc bkd	ar × A B	Pro y B C D	ccurrenceOf	rty Ta oid ab1 bc1 bd1	iid akb bkc bkd	:color :	oid ab1 bc1 bd1	color red blue blue	:dtype	WAYS oid ab1 bc1 bd1	dtype 
	RDFnPropertynamexyakbABbkcBCbkdCDckdCD	Tables: Definition Definitio	e color red blue blue green	ertype dtype	nceOf name akb bkc bkd ckd		RI smoux:	DF-st iid akb bkc bkd ckd	ar × A B B C	Pro y B C D D	:occurrenceOf	rty Ta oid ab1 bc1 bd1 cd1	ables: iid akb bkc bkd ckd	:color	oid ab1 bc1 bd1 cd1	color red blue blue green	:dtype	WAYS oid ab1 bc1 bd1 cd1	dtype 

# **Multi-Edge Handling Costs:** RDFn Named Triples vs. RDF-star Alternatives Data Size, Burden on Creator, Query Complexity/Efficiency, Future-Proof?

When a single property occurring in N triples transitions to multi-edge(s) adding m extra triples ...

<ul> <li>N ← #distinct triples using given property</li> <li>m ← #extra triples for the multi-edge(s)</li> <li>N can be high, m usually small</li> </ul>	RDFn Named Triples	:occurrenceOf, for extra edges in multi-edges	:occurrenceOf, for all edges in multi-edges	:occurrenceOf, for all edges
<b>Data Size:</b> # core triples (not counting "statement about statement" triples)	N + m	N + m	(N + m + 1) to (N + 2*m)	2*N + m
Burden (on data creator): # custom IRIs that the data creator has to provide)		(Da	ata Size – N)	
<b>COUNT Query Complexity/Efficiency:</b> #patterns / Simple, UNION, OPTIONAL?	1/Simple	2/UNION	2/OPTIONAL	2/Simple
Future-Proof?: Pre-transition COUNT query still works?	YES	NO	NO	YES

... and the overhead increases (additively) when multiple properties transition to multi-edges!



