

階層構造データの高速集計の原理・説明図

Hyper Tree(HT)技術のコア部分

[出典:特許公報・特許第3230677号＝平成13年11月19日発行]

図1 入力データ列: 上から順に入ってくる

(b)	<table border="1"><tr><th>店舗</th><th>商品</th><th>注文個数</th></tr><tr><td>A1</td><td>B2</td><td>N1</td></tr></table> ←レコード1(1番目のデータ)	店舗	商品	注文個数	A1	B2	N1
店舗	商品	注文個数					
A1	B2	N1					
(c)	<table border="1"><tr><th>店舗</th><th>商品</th><th>注文個数</th></tr><tr><td>A1</td><td>B1</td><td>N2</td></tr></table> ←レコード2(2番目のデータ)	店舗	商品	注文個数	A1	B1	N2
店舗	商品	注文個数					
A1	B1	N2					
(d)	<table border="1"><tr><th>店舗</th><th>商品</th><th>注文個数</th></tr><tr><td>A1</td><td>B3</td><td>N3</td></tr></table> ←レコード3(3番目のデータ)	店舗	商品	注文個数	A1	B3	N3
店舗	商品	注文個数					
A1	B3	N3					
(e)	<table border="1"><tr><th>店舗</th><th>商品</th><th>注文個数</th></tr><tr><td>A2</td><td>B1</td><td>N4</td></tr></table> ←レコード4(4番目のデータ)	店舗	商品	注文個数	A2	B1	N4
店舗	商品	注文個数					
A2	B1	N4					

図3 階層構造でデータを管理するインデックス・テーブル
(これで、高速集計のアイデアを実現)

	ノード	レベル	L Link	C Link	R Link	値	
(a)	1	ROOT	0	4	-1	-1	N1+N2+N3+N4
(b)	2	B2	1	3	1	6	N1
(c)	3	A1	2	-1	2	-1	N1
(d)	4	B1	1	5	1	2	N2+N4
(e)	5	A1	2	-1	4	8	N2
(d)	6	B3	1	7	1	-1	N3
(e)	7	A1	2	-1	6	-1	N3
(e)	8	A2	2	-1	4	-1	N4

表中の-1は、下位なし、上位なし、同位の下なしを示す。

L Link: 下位を指すリンク

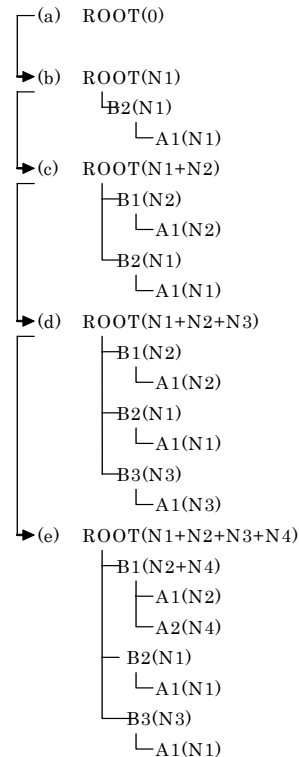
C Link: 上位を指すリンク

R Link: 同位の、すぐ下の弟妹を指すリンク

(この3つのリンクで、各ノードに関する階層関係を動的に管理)

値: 各ノードの傘下のデータの合計値を記録

〔図2〕 入力データを商品(Bi)をキーに階層構造化しながら、商品別・店舗別に逐次集計する。



← 入力データが入る度に (矢印の上から順に) ノード間の関係を更新しながら、ノード毎に逐次集計を行なう。