STUDIA INFORMATICA 2007
Volume 28 Number 3A (72)

Marcin GORAWSKI, Marcin BUGDOL
Politechnika Śląska, Instytut Informatyki

KASKADOWE OPERACJE ECOLAP

Streszczenie. W artykule przestawione zostały nowe definicje operacji kaskadowych ECOLAP z użyciem wyrażeń algebry relacji. Operacje te zostały zdefiniowane dla przestrzennych hurtowni danych o schemacie rozszerzonej gwiazdy kaskadowej na bazie operacji COLAP dla schematu gwiazdy kaskadowej.

Słowa kluczowe: przestrzenne hurtownie danych, pojedyncza gwiazda, płatek śniegu, schemat rozszerzonej gwiazdy kaskadowej, COLAP.

CASCADED ECOLAP OPERATIONS

Summary. The paper proposes the new definitions of *Expanded Cascaded OLAP* (ECOLAP) operations by using the relation algebra. The operations have been defined for the extended cascaded star schema in spatial data warehouse SDW basing on the existing definitions of the COLAP operations for the cascaded star schema. Moreover, they support extensions presented in the new logical model.

Keywords: spatial data warehouses, single star, snowflake, extended cascaded star schema, COLAP.

1. Wstęp

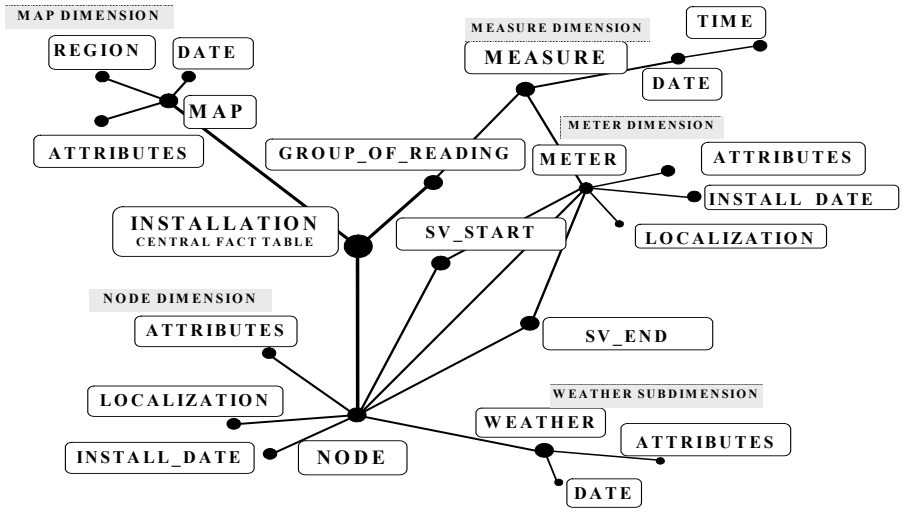
Rosnąca potrzeba tworzenia satelitarnych map geograficznych i ich użycie dla lokalizacji milionów obiektów wywołała dynamiczny rozwój bardzo dużych repozytoriów danych przestrzennych zorientowanych na zaawansowane analizy, nazywanych systemami przestrzennych hurtowni danych ((ang. *Spatial Data Warehouse*_SDW) [1]. Stąd konieczność definiowania nowych schematów logicznych SDW oraz modeli analitycznego przetwarzania typu on-line OLAP zarówno dla danych przestrzennych, jak i danych płaskich [2].

Dotychczasowe struktury logiczne modeli OLAP tj.: schemat gwiazdy, schemat płatka śniegu oraz schemat konstelacji faktów, nie są w stanie zamodelować optymalnie danych

przestrzennych ze względu na ich hiperwymiarowość i wielotematyczność. Wymienione schematy nadają się do reprezentacji danych zorientowanych jednoprocesowo, a skojarzone operacje OLAP-owe są wykonywane odpowiednio wzdłuż hierarchii wymiarów. Zgodnie z tym, wykorzystanie schematu gwiazdy czy płatka śniegu, przedstawiającego pojedynczy proces (temat), powoduje, że złożone zapytanie do SDW odnoszące się do wielu równoczesnych procesów opisanych różnymi poziomami wymiarów musi zostać zdefiniowane jako kilka odrębnych zapytań, których wyniki łączy użytkownik [3].

W pracy [2] zaprezentowano sformalizowany logiczny model ROLAP, nazywany gwiazdą kaskadową (ang. *Cascaded Star*), który pozwala modelować hiperwymiarowe dane i wielotematyczne procesy wspomagania podejmowania decyzji. Pomimo że ten schemat gwiazdy kaskadowej w znaczącym stopniu rozszerzył możliwości modelowania schematów logicznych na potrzeby SDW, to jednak zdarzają się sytuacje, w których model ten jest niewystarczający do opisu przestrzennych, wielotematycznych zjawisk. Przykładem jest system przestrzennej hurtowni danych telemetrycznych SDW(t), nad którym pracuje Zespół Algorytmów, Programowania i Systemów Autonomicznych w Zakładzie Teorii Informatyki Instytutu Informatyki Politechniki Śląskiej [4,5,6].

W pracy [3] zebrano reprezentatywne schematy logiczne systemów SDW(t). Takim przykładem jest schemat wielowersyjnej gwiazdy kaskadowej w systemie SDW(t) zorientowanej na pomiar (rys. 1).



Rys. 1. Schemat wielowersyjnej gwiazdy kaskadowej zorientowanej na pomiar
Fig. 1. Multiversioned cascaded star oriented on measure schema

Składa się on z centralnej tabeli faktów *INSTALLATION* oraz pięciu wymiarów *NODE, METER, MEASURE, MAP* oraz *WEATHER*. W schemacie tym znajdują się również dwie tabele *SV_START* oraz *SV_STOP*, które zapewniają wielowersyjność schematu. Przechowują one informacje na temat przedziałów ważności liczników i węzłów. Właśnie wielowersyj-

ność tego schematu odróżnia go od zdefiniowanego w pracy [1] schematu gwiazdy kaskadowej wraz z operacjami COLAP (ang. *Cascaded OLAP).*

W pracy [3] zaprezentowano nowy, sformalizowany logiczny model ROLAP, nazywany rozszerzoną gwiazdą kaskadową (ang. *Expanded Cascaded Star,* $ES^{C}_{\{P-t\}}$), który pozwala modelować hiperwymiarowe dane i wielotematyczne procesy wspomagania podejmowania decyzji. Zdefiniowano tam rodzinę $ES^{C}_{\{P-t\}}$, gdzie *P* jest stopniem schematu określanym typem i rodzajem podgwiazd oraz ich wielowersyjnością, natomiast *t* jest liczbą jednocześnie odwzorowywanych tematów.

W zależności od stopnia rozbudowy gwiazdy kaskadowej rozróżniamy kilka jej postaci, m.in:

- **Rozszerzona gwiazda kaskadowa I stopnia** $ES^{C}_{\{I-t\}}$ charakteryzuje gwiazdę kaskadową, uwzględniającą wyłącznie schematy pojedynczych gwiazd (zgodnie z definicją schematu gwiazdy kaskadowej i pojedynczej wg pracy [2]).
- **Rozszerzona gwiazda kaskadowa II stopnia** $ES^{C}_{\{II-t\}}$ to rozszerzenie gwiazdy kaskadowej I stopnia o schemat płatka śniegu (zgodnie z definicją schematu gwiazdy kaskadowej, pojedynczej oraz płatka śniegu).

Gwiazdę *P-tego* stopnia $ES^{C}_{\{P-t\}}$ definiuje się jako gwiazdę (*P-1*) stopnia z nowym komponentem. Definicja rozszerzonej gwiazdy kaskadowa II stopnia dla t=1 za pracą [3] ma postać jak niżej.

Definicja 1. Rozszerzona gwiazda kaskadowa II stopnia dla t=1 [3]

Schemat rozszerzonej gwiazdy kaskadowej II stopnia dla k=1, $ES^{C}_{\{II-1\}}$ definiowany jest jako $ES^{C}_{\{II-1\}}$ = (SS, H^C, T^C, L^C), gdzie:

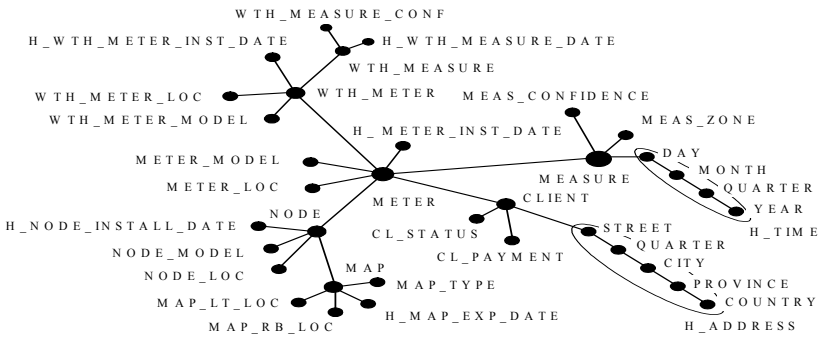
1. SS jest zbiorem wymiarów kaskadowych – schematów danych, takich że $S_i \in$ SS jest:
   a) schematem pojedynczej gwiazdy S^S lub
   b) schematem płatka śniegu S^F, lub
   c) schematem $ES^{C}_{\{II-1\}}$.

2. H^C jest zbiorem wymiarów hierarchicznych ES^C. Każdy wymiar hierarchiczny $H_i \in H^C$ ma określony stopień L_i, definiowany jako liczba poziomów hierarchii w schemacie S^F.
3. $T^C = [V^C, PK_{SR}, PK_H]$ jest kaskadową tablicą faktów, gdzie V^C jest zbiorem centralnych pomiarów faktów, $V^C = \{V_{SF} \cup V_S\}$. $PK_{SR} = \{SRi\}$ jest zbiorem sztucznych kluczy wymiarów kaskadowych $S_i \mid S_i \in SS$, a $PK_H = \{PK_i\}$ jest zbiorem kluczy głównych wymiarów hierarchicznych $H_i \mid H_i \in H^C$. Kluczem głównym PK_i wymiaru H_i jest klucz główny tabeli h_0: $PK_{h_{i0}}$.

Schemat ES^C może zawierać zarówno zagnieżdżone schematy gwiazdy, jak i zagnieżdżone schematy płatka śniegu. Rozszerzenie to pozwala na przeprowadzenie normalizacji złożonych tabel wymiarów.

2. Operacje ECOLAP dla rozszerzonej gwiazdy kaskadowej

Wykorzystanie nowej struktury $ES^C_{\{P-t\}}$ w analizie OLAP wymaga zdefiniowania odpowiednich operacji, które tę analizę wspomogą. Poniżej zdefiniowaliśmy operację ECOLAP (ang. *Expanded Cascaded OLAP)* dla rozszerzonej gwiazdy kaskadowej o schemacie $ES^C_{\{II-1\}}$ na przykładzie schematu $ES^C_{\{II-1\}}SDW(t)$. Przedstawia to rys. 2 (opisany szczegółowo w [3]).



Rys. 2. Schemat $ES^C_{\{II-1\}}SDW(t)$ zorientowany na pomiar

Fig. 2. $ES^C_{\{II-1\}}SDW(t)$ oriented on measure schema

Operacje COLAP [2] nie uwzględniają schematu płatka śniegu. Z tego powodu dla operacji ECOLAP zdefiniowano najpierw operacje przechodzenia wzdłuż danej hierarchii dla wymiaru w postaci schematu płatka śniegu.

2.1. Hierarchiczne złączenie

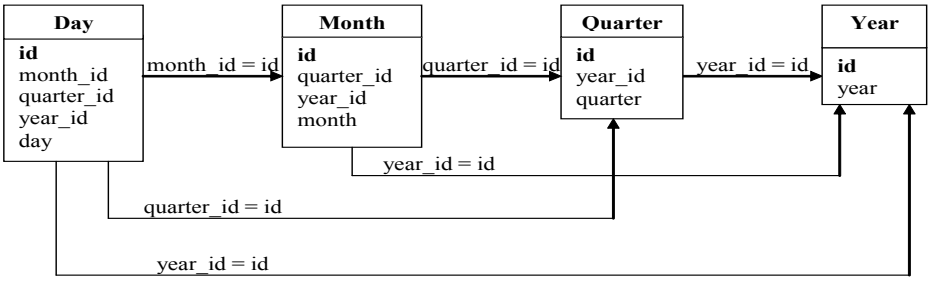
Hierarchiczne złączenie (ang. *Hierarchical-join*) jest to operacja wykonująca złączenia dla danej hierarchii. Złączenie wykonywane jest od poziomu najniższego do poziomu

wyznaczonego przez drugi z parametrów operacji. Operację tę definiujemy w następujący sposób:

$$\textit{Hierachical-Join}(H_i, h_i) = \pi_{R_H}(\sigma_{(h=C)}(h_1 \bowtie \ldots \bowtie h_i)),$$

gdzie: pierwszy parametr H_i określa hierarchię ze zbioru hierarchii danego schematu płatka śniegu, z kolei drugi parametr h_i jest tablicą, do której należy wykonywać złączenie. Jeśli nie podamy tablicy h_i, to operacja zwróci zbiór pusty. R_H jest zbiorem atrybutów projekcji, h zbiorem atrybutów w obrębie hierarchii, natomiast C zbiorem warunków selekcji, które muszą zostać spełnione.

Rysunek 3 prezentuje przykładową strukturę hierarchicznego wymiaru czasu.



Rys. 3. Hierarchiczny wymiar czasu *H_TIME* schematu $ES^{C}_{\{II-1\}}SDW(t)$

Fig. 3. Hierarchical time dimension *H_TIME* for $ES^{C}_{\{II-1\}}SDW(t)$ schema

Dla tej struktury zaprezentujemy przykład użycia operacji *Hierarchical-Join*.

Zapytanie 1

Podać identyfikatory dni, które należą do trzeciego kwartału roku 2007.

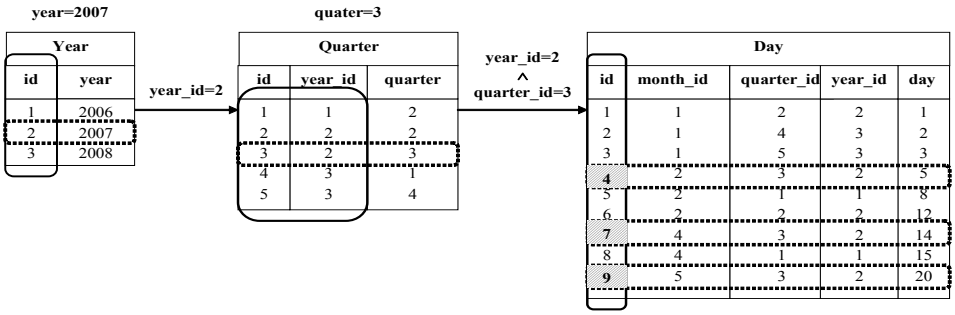
Aby uzyskać potrzebne wyniki, należy najpierw pobrać identyfikator roku oraz kwartału. W tym celu należy wykonać złączenie hierarchiczne do poziomu tabeli *YEAR*. Zapis w algebrze relacji operacji wykonującej powyższe zadanie wygląda następująco:

$$\textit{Hierarchical-Join}(\texttt{H\_TIME}, \{\texttt{YEAR}\}) = \pi_{day.id}(\sigma_{(quater=3 \wedge year=2007)}(\texttt{DAY} \bowtie \texttt{QUARTER} \bowtie \texttt{YEAR})).$$

Na rys. 4 zaprezentowano sposób wykonania operacji na przykładowych danych. Użyte zostały następujące oznaczenia:

- − atrybuty projekcji dla danej tabeli lub kolumny, po których następuje złączenie,
- − wybrane wiersze z danej tabeli (spełniające warunki selekcji),
- − ostateczny wynik zapytania,
- − złączenie tabel poprzez atrybuty znajdujące się w opisie oznaczenia i kolumny zaznaczone w tabeli, z której strzałka wychodzi.

W niektórych przykładach z powodu rozmiaru tabeli, liczba kolumn została ograniczona do niezbędnego minimum.



Rys. 4. Przykład wykonania operacji SHJ dla wymiaru czasu *H_TIME*
Fig. 4. Example of execution SHJ operation for time dimension *H_TIME*

Wynikiem zapytania jest następujący zbiór R = **{4, 7, 9}**.

2.2. Hierarchiczne złączenie dla całego schematu płatka śniegu

Hierarchiczne złączenie dla całego schematu płatka śniegu (ang. *Schema-Hierarchical--Join* w skrócie *SHJ*) to operacja, która przyjmuje jako parametry zbiór wymiarów hierarchicznych oraz zbiór tablic, do których należy wykonać złączenie. Na tej podstawie generuje zbiór wymiarów hierarchicznych po wykonaniu złączeń.

$$SHJ : H \times I \rightarrow N \; ; \; SHJ(H_i, h_i) = Hierarchical\text{-}Join(H_i, h_i),$$

gdzie: $H = \{H_i\}$ – zbiór wymiarów hierarchicznych, $I = \{h_i\}$ – zbiór tablic, które dla danej hierarchii H_i określają graniczny poziom złączenia h_i w ramach tej hierarchii, N – zbiór wymiarów hierarchicznych po wykonaniu złączeń.

2.3. ePrzechodzenie

ePrzechodzenie (ang. *expanded traverse, eTraverse*) umożliwia podstawowe operacje OLAP w ramach pojedynczego wymiaru S^S lub S^F dla pewnego poziomu gwiazdy $M = k$, $k > 0$. Składa się z operatorów relacji (*SPJ*) na prostych atrybutach oraz operacji *SHJ* dla wymiarów hierarchicznych. W algebrze relacji można zapisać tę operację jako:

$$eTraverse(S,I) = \pi_{R_S} \; (\sigma_{(d=C)}(S \bowtie D \bowtie SHJ(H, I))),$$

gdzie: S jest schematem S^S lub S^F, D jest zbiorem wymiarów płaskich, H jest zbiorem wymiarów hierarchicznych, I jest zbiorem tablic, które dla danej hierarchii $H_i \in H$ określają graniczny poziom złączenia h_i w ramach tej hierarchii, d jest zbiorem atrybutów dla D oraz H, C jest zbiorem warunków selekcji, natomiast R_s jest zbiorem atrybutów projekcji w ramach S.

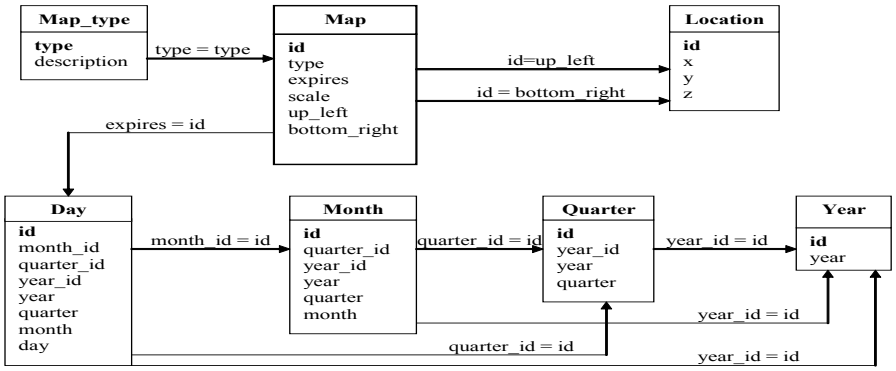
Poniżej przedstawiono przykładowe zapytanie, które dowodzi poprawności przedstawionej definicji.

Zapytanie 2

Wyszukać identyfikatory, nazwę typów oraz skalę map, których data ważności upływa w 2008 roku.

Zapytanie dotyczy tylko pojedynczego podwymiaru o schemacie płatka śniegu. Struktura fizyczna tego podwymiaru została przedstawiona na rys. 5. Ponieważ hierarchia czasu zawiera na każdym poziomie wszystkie pola z wyższych poziomów (schemat płatka śniegu typu 3 [3]), nie trzeba wykonywać żadnych złączeń w ramach hierarchii. Jest to szczególny przypadek, który został zaprezentowany, aby udowodnić, że operacja SHJ jest poprawna również w takiej sytuacji. Zapytanie to zapisane za pomocą wyrażeń algebry relacji wygląda następująco:

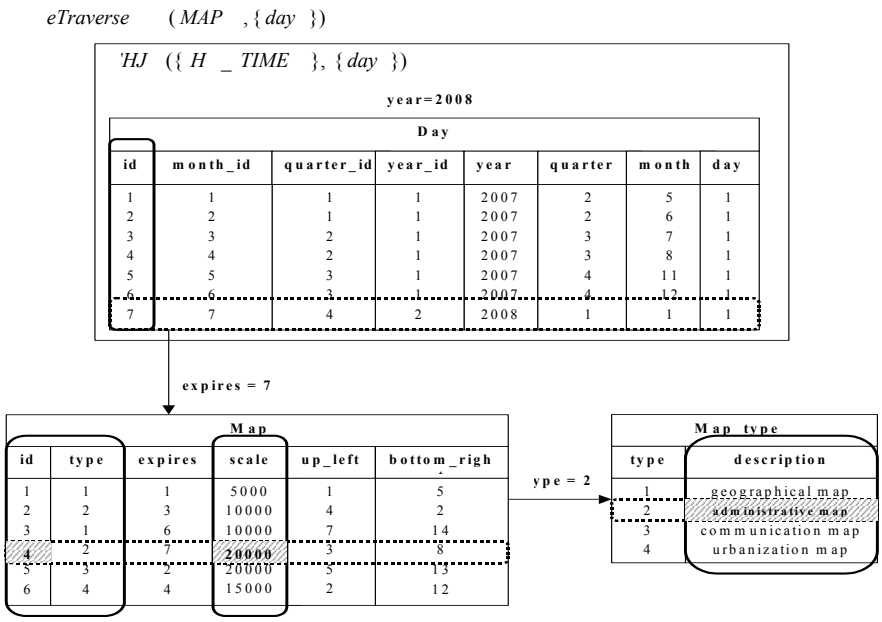
$$eTraverse(MAP,\{day\}) = \pi_{map.id,map\_type.description,map.scale}(\sigma_{(year=2008)}(MAP \rhd\lhd \{MAP\_TYPE\} \rhd\lhd SHJ(\{H\_TIME\},\{day\})))$$



Rys. 5. Wymiar *MAP* schematu $ES^{C}_{\{II-1\}}SDW(t)$

Fig. 5. *MAP* dimension for $ES^{C}_{\{II-1\}}SDW(t)$ schema

Operacja $SHJ(\{H\_TIME\},\{day\})$ zwraca jako wynik tabelę *DAY* i ona jest łączona z wynikiem złączenia tabel *MAP* oraz *MAP_TYPE*. Poniżej (rys. 6) został zaprezentowany sposób wykonania zapytania na przykładowych tabelach danych.



Rys. 6. Przykład wykonania operacji *eTraverse* dla zapytania 2
Fig. 6. Example of execution of *eTraverse* operation for query 2

Równoważne zapytanie SQL wygląda następująco:

```
SELECT m.scale, mt.description
FROM map m, map_type mt,day d
WHERE m.EXPIRES = d.ID AND m.TYPE = mt.TYPE AND d.YEAR = 2008
```

Wynikiem zapytania jest zbiór krotek:

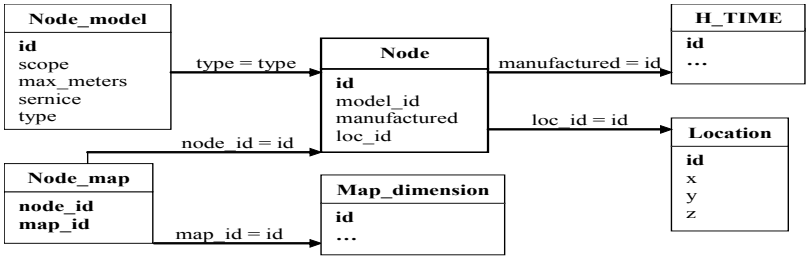
Result		
id	**scale**	**description**
4	20000	administrative map

2.4. eDekompozycja

Opis operacji dekompozycji (ang. *decompose*) został umieszczony w [1]. Poniżej uzupełnienio tę definicję o obsługę wymiarów hierarchicznych, zawartych w schematach płatka śniegu. Operację eDekompozycji (ang. *expanded decompose, eDecompose*) można zapisać w algebrze relacji jako:

$$eDecompose(Q, P, I) = \pi_{R_Q}(\sigma_{(d=C)}(Q \rhd\lhd eTraverse(P, I))),$$

gdzie: P jest schematem S^S lub S^F posiadającym zbiór wymiarów hierarchicznych H i będącym pojedynczym wymiarem dla Q, który z kolei jest schematem rozszerzonej gwiazdy kaskadowej $ES^C_{\{H-1\}}$, I jest zbiorem tablic, które dla danej hierarchii $H_i \in H$ określają graniczny poziom złączenia h_i w ramach tej hierarchii, d jest zbiorem atrybutów dla P, C jest zbiorem warunków selekcji, natomiast R_p jest zbiorem atrybutów projekcji w ramach Q.



Rys. 7. Kaskadowy wymiar *NODE* schematu $ES^{C}_{\{II-1\}}SDW(t)$

Fig. 7. Cascaded dimension *NODE* for $ES^{C}_{\{II-1\}}SDW(t)$ schema

Rysunek 7 przedstawia schemat kaskadowego wymiaru *NODE*. Dla uproszczenia rysunku hierarchię czasu oraz wymiar *MAP* zaznaczono symbolicznie. Poniżej, w celu wykazania poprawności definicji, zaprezentowano przykładowe zapytanie.

Zapytanie 3

Wyszukać informacje o węzłach (identyfikator oraz identyfikator modelu węzła) posiadających mapy obszarów, których data ważności upływa w 2008 roku.

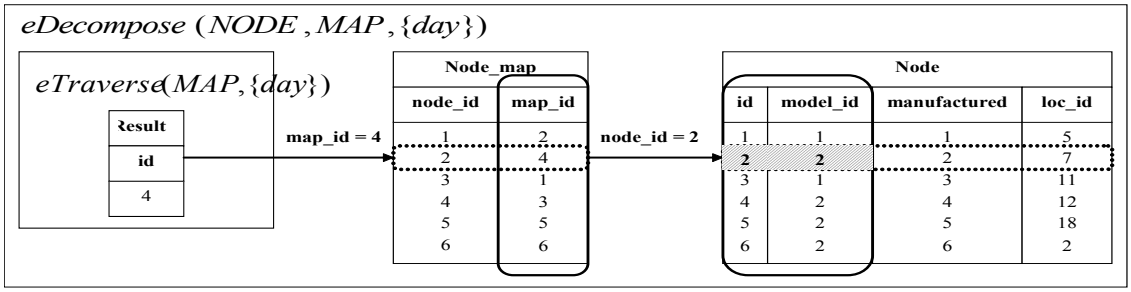
Zapytanie odnosi się zarówno do wymiaru kaskadowego *NODE,* jak i do wymiaru płatka śniegu *MAP*. Zapytanie to zapisane za pomocą wyrażeń algebry relacji przedstawia się następująco:

$$eDecompose(NODE, MAP, \{day\}) = \pi_{node.id,node.\mathrm{mod}el\_id}(\sigma_{(year=2008)}(NODE \rhd\lhd eTraverse(MAP, \{day\})))$$

Rozwijając operację *eTraverse* oraz uwzględniając fakt, że połączenie tabel *NODE* oraz *MAP* może nastąpić tylko poprzez tabelę *MAP_NODE,* należy zapisać:

$$eDecompose(NODE, MAP, \{day\}) = \pi_{node.id,node.\mathrm{mod}el\_id}(NODE \rhd\lhd MAP\_NODE \rhd\lhd (\pi_{map.id}(\sigma_{(year=2008)}(MAP \rhd\lhd SHJ(\{H\_TIME\}, \{day\})))))$$

Tak jak w poprzednim zapytaniu, operator *SHJ* zwraca jako wynik tabelę *DAY*, która jest łączona z tabelą *MAP*. Wynikiem operacji *eTraverse* jest tabela identyfikatorów map, które spełniają warunek selekcji. Na tej podstawie wybierane są węzły znajdujące się na tych mapach. Poniżej (rys. 8) został pokazany sposób wykonania powyższego zapytania na przykładowych danych. Dla uproszczenia operacja *eTraverse* zostanie pominięta i wykorzystano tylko tablicę wyników, która jest identyczna z zaprezentowaną przy definicji poprzedniej operacji:



Rys. 8. Przykład wykonania operacji *eDecompose*
Fig. 8. Example of execution of *eDecompose* operation

Wynikiem zapytania 3 jest krotka o postaci:

Result	
id	**model_id**
2	2

Równoważne zapytanie SQL wygląda następująco:

```
SELECT n.id, n.model_id
FROM node n, map_node mn, map m, day d
WHERE n.id = mn.node_id AND mn.map_id = m.ID AND m.expires = d.id
AND d.year = 2008
```

2.5. eSkok

Operację skoku (ang. *jump*) zdefiniowano w pracy [2]. Definicja ta nie jest kompletna. Operacja złączenia dwóch gwiazd, pomiędzy którymi nie ma żadnej bezpośredniej relacji, skutkuje zbiorem wynikowym, będącym iloczynem kartezjańskim obu gwiazd. Taki zbiór, będzie zawierał ogromną liczbę rekordów. Z tego powodu należy przyjąć definicję, że jest to złączenie pomiędzy dwoma gwiazdami z użyciem dodatkowych tabel faktów zawierających odpowiednie relacje. Na przykład, dla schematu $ES^{C}_{\{II-1\}}$ przedstawionego powyżej, pomiędzy wymiarem *CLIENT* a wymiarem *MAP* nie istnieją żadne bezpośrednie relacje. W tym przypadku należy dokonać złączenia tabeli *CLIENT* z *METER*, a następnie z tabelą *NODE* i w końcu z tabelą *MAP*. Operację eSkoku (ang. *expanded jump, eJump*) z uwzględnieniem wymiarów hierarchicznych w postaci algebry relacji można przedstawić następująco:

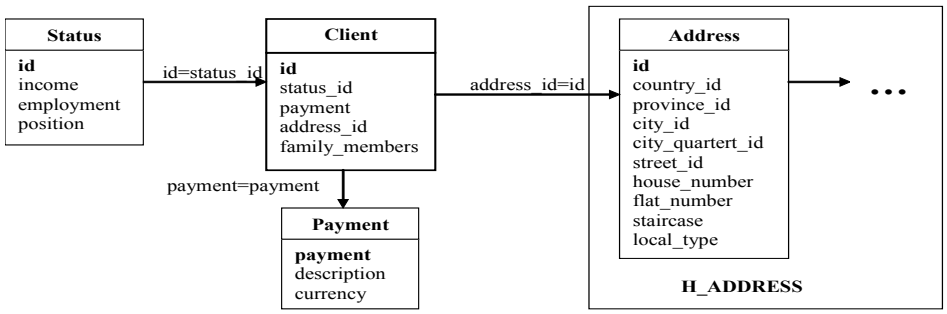
eJump(Si, Sj, Ii, Ij) = $\pi_{R_{i,j}}$ (eTraverse(Si, Ii) $\bowtie$ eTraverse(Sj, Ij))),

gdzie: S_i oraz S_j wskazują źródłowy i docelowy schemat S^S lub S^F posiadający zbiory wymiarów hierarchicznych H_{S_i} i H_{Sj}, $R_{i,j}$ jest zbiorem atrybutów relacji dla operacji projekcji z S_i oraz S_j, natomiast I_i i I_j są zbiorami tablic, które dla każdej z hierarchii $H_i \in H_{S_i}$ oraz $H_j \in H_{Sj}$ określają graniczny poziom złączenia h_i i h_j w ramach tych hierarchii.

Rysunek 9 przedstawia wymiar *CLIENT*. Wymiar hierarchiczny *H_ADDRESS* został pominięty, ponieważ nie będzie on wykorzystany w przykładowym zapytaniu, a rysunek jest bardziej czytelny. Zostanie przedstawione kolejne przykładowe zapytanie, którego wykonanie będzie wymagało użycia operacji *eJump*.

Zapytanie 4

Podać wszystkich klientów, których dochód roczny wynosi ponad 30000 i znajdujących się w obszarze, którego mapy wygasają w 2008 roku.



Rys. 9. Wymiar *CLIENT* schematu $ES^{C}_{\{II-1\}}SDW(t)$

Fig. 9. *CLIENT* dimension for $ES^{C}_{\{II-1\}}SDW(t)$ schema

Zapytanie odwołuje się do wymiarów *CLIENT* oraz *MAP*. Jednak, aby połączyć oba te wymiary, należy połączyć ze sobą tabele *CLIENT, METER, NODE* oraz *MAP*. Dodatkowo, w schemacie *NODE* należy odwołać się do hierarchii *H_TIME*. Hierarchia czasu jest schematem płatka śniegu typu 3, dlatego też nie trzeba wykonywać dodatkowych złączeń. W wymiarze *CLIENT* nie ma odwołań do hierarchii adresu. Powyższe zapytanie można zapisać za pomocą wyrażeń algebry relacji:

$$eJump(CLIENT, MAP, \{\}, \{day\}) = \pi_{client.id}(eTraverse(CLIENT, \{\}) \rhd\lhd$$

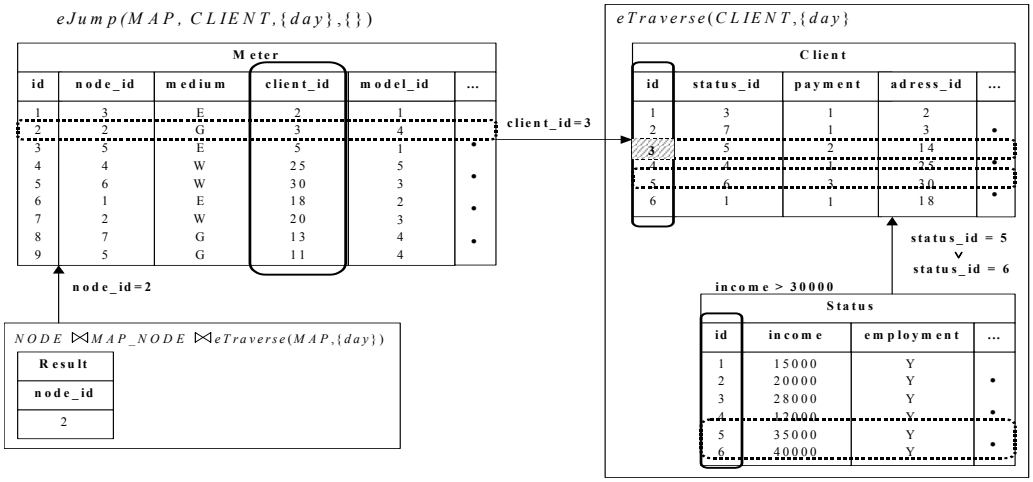
$$eTraverse\,(MAP, \{day\})$$

Rozwinięcie powyższego wyrażenie z uwzględnieniem wszystkich tabel oraz hierarchii czasu przedstawia się następująco:

$$eJump(CLIENT, MAP, \{\}, \{day\}) = \pi_{client.id}(CLIENT \rhd\lhd SHJ(\{H\_ADDRESS\}, \{\}) \rhd\lhd$$
$$METER \rhd\lhd NODE \rhd\lhd MAP\_NODE \rhd\lhd (\pi_{map.id}(\sigma_{(year=2008)}(MAP \rhd\lhd$$
$$SHJ(\{H\_TIME\}, \{day\}))))$$

Operacja *SHJ* dla hierarchii *H_ADDRESS* wymiaru *CLIENT* zwraca zbiór pusty, więc można pominąć tę operację w zapisie algebry relacji. Natomiast operacja *SHJ* dla hierarchii *H_TIME* wymiaru *MAP* zwraca jako wynik tabelę *DAY*.

Rysunek 10 przedstawia wykonanie zapytania 4 na przykładowych tabelach danych. W celu uproszczenia zapisu wykorzystano wynik poprzedniego zapytania, tzn. identyfikator węzła, do którego należą mapy, których ważność upływa w 2008 roku.



Rys. 10. Przykład wykonania operacji *eJump*
Fig. 10. Example of execution of *eJump* operation

2.6. eKaskadowe zwijanie (eCascaded-roll-up)

Operacja zwijania polega na zmniejszeniu szczegółowości prezentowanych danych, w czasie której może dochodzić do pomijania całych wymiarów. W przypadku schematu pojedynczej gwiazdy oraz płatka śniegu operację tę można wykonać za pomocą operacji *eTraverse(S,I)*, zmieniając w kolejnych etapach zbiór warunków selekcji oraz zbiór atrybutów projekcji. Natomiast w przypadku schematu rozszerzonej gwiazdy kaskadowej ruch odbywa się wzdłuż wymiarów, które poza najwyższym poziomem są gwiazdami kaskadowymi. Punktem wyjściowym jest pojedyncza gwiazda S^S lub płatek śniegu S^F. Wykonując operację *eTraverse*, otrzymuje się zbiór atrybutów ze zbioru pojedynczego schematu. Następnie ma miejsce przejście na niższy poziom do gwiazdy rodzica, która jest schematem $ES^C_{\{II-1\}}$. Aby połączyć oba poziomy (pojedynczy schemat i $ES^C_{\{II-1\}}$), należy wykonać operację *eDecompose*. W ten sposób iteracyjnie osiągnięty jest najniższy poziom M=0. Za pomocą wyrażeń algebry relacji proces ten można zapisać:

$$eCascaded\_roll\_up(Q, P, I_q, I_p) = eDecompose(Q, eTraverse(P, I_p), I_q),$$

gdzie: Q jest schematem $ES^C_{\{II-1\}}$, P jest schematem pojedynczej gwiazdy S^S lub schematem płatka śniegu S^{SF} posiadającego zbiór wymiarów hierarchicznych H, natomiast I_p i I_q są zbiorami tablic, które dla danych hierarchii $H_p \in H$ oraz $H_q \in H$ określają graniczny poziom złączenia h_p i h_q w ramach tych hierarchii.

2.7. eKaskadowe rozwijanie (eCascaded drill-down)

Operacja ta jest operacją odwrotną do kaskadowego zwijania. Polega ona na zwiększeniu ziarnistości przedstawionych danych, a w szczególnym przypadku pojawienie się nowych wymiarów. Ruch odbywa się od centrum gwiazdy do jej najwyższego poziomu, dodając po

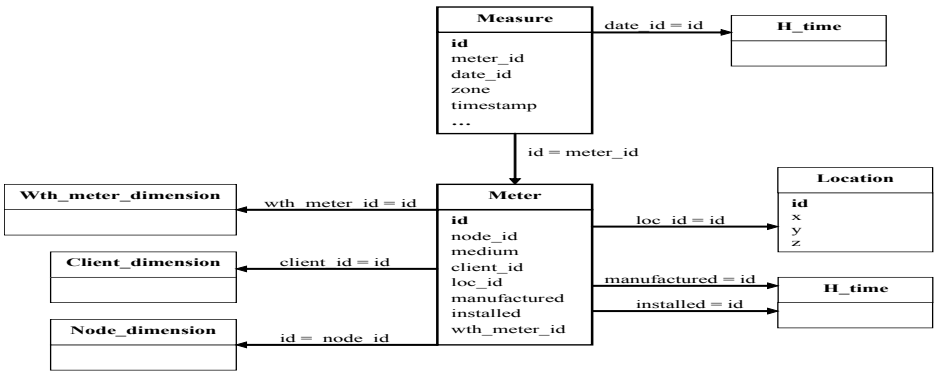
drodze dodatkowe podgwiazdy lub wymiary. Rozpoczynając w gwieździe centralnej i wykonując kolejne operacje *eDecompose*, osiągnięty zostaje poziom pojedynczej gwiazdy, dla której wykonywana jest operacja *eTraverse*. Zapis w postaci algebry relacji wygląda następująco:

$$eCascaded\_drill\_down(Q, P, I_{pq}, I_q) = eTraverse(eDecompose(Q, P, I_{pq}), I_q),$$

gdzie: Q jest schematem $ES^{C}_{\{II-1\}}$, P jest schematem pojedynczej gwiazdy S^{S} lub schematem płatka śniegu S^{SF} posiadającego zbiór wymiarów hierarchicznych H, natomiast I_{pq} i I_q są zbiorami tablic, które dla danych zbiorów hierarchii $H_{pq} \in H$ oraz $H_q \in H$, określają graniczny poziom złączenia h_p i h_q w ramach tych hierarchii.

Poniżej została zaprezentowana operacja *eCascaded drill-down* z wykorzystaniem hierarchii zbudowanej z wymiarów *MEASURE, METER* oraz *CLIENT* (rys. 11).

Na początku zostało pokazane zapytanie generujące zestawienie pomiarów (wartość oraz data pomiaru). Następnie, poruszając się wzdłuż hierarchii wymiarów *METER* oraz *CLIENT*, wykonana została operacja *drill-down*, co poszerzyło zakres prezentowanych danych.



Rys. 11. Kaskadowy wymiar *MEASURE* oraz *METER*
Fig. 11. Cascaded dimension *MEASURE* and *METER*

Poziom I (najmniej szczegółowy) – identyfikator, data i wartość pomiaru

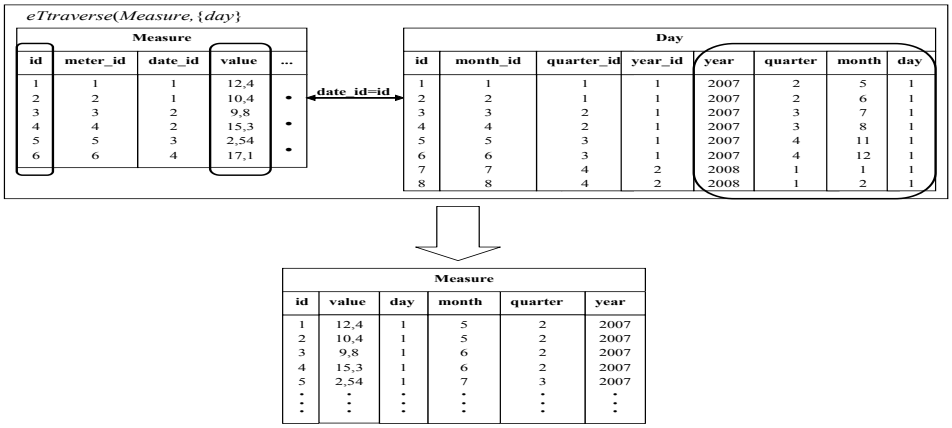
Potrzebne informacje przechowywane są w tabeli faktów wymiaru kaskadowego *MEASURE* oraz jego wymiarze hierarchicznym *H_TIME*. Ponieważ nie ma potrzeby odwołania się do innych wymiarów kaskadowych, więc operacja *drill-down* upraszcza się do operacji *eTraverse*. W algebrze relacji zapytanie to przedstawia się następująco:

$$eTraverse(MEASURE, \{day\}) = \pi_{measure.value, day.day, day.month...}(MEASURE \triangleright\!\triangleleft SHJ(\{H\_TIME\}, \{day\}))$$

Poniżej (rys. 12) zaprezentowano sposób wykonania tej operacji. Opisy powyżej strzałek oznaczają kolumny, po których następuje złączenie.

Równoważne zapytanie SQL:

```
SELECT m.value, d.day, day.month, day.year
FROM meter mt, day d
WHERE m.date_id = d.id
```



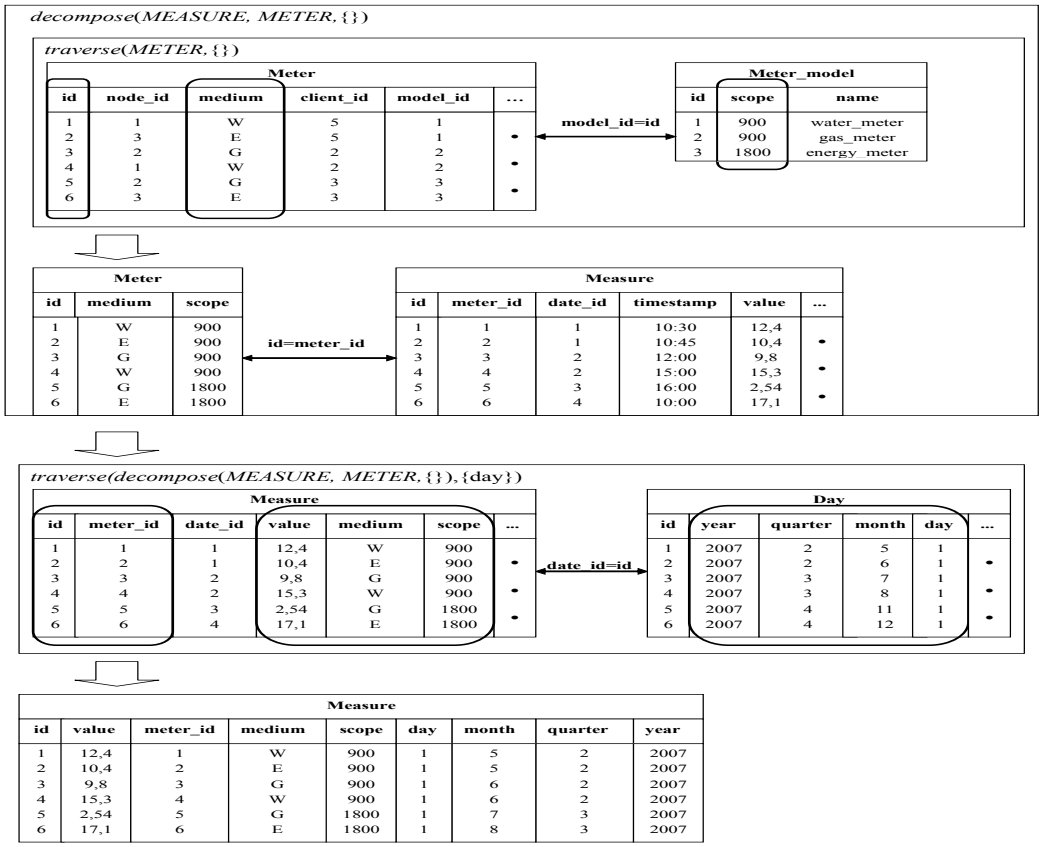
Rys. 12. Pierwszy poziom operacji *eCascaded drill-down*
Fig. 12. First level of *eCascaded drill-down* operation

Poziom II – data, wartość pomiaru, identyfikator licznika, jego zakres i mierzone medium

W tym przypadku konieczne jest wykorzystanie również informacji zawartych w wymiarze kaskadowym *METER*. Dlatego należy najpierw dokonać dekompozycji wymiaru *MEASURE* poprzez wymiar *METER*, a następnie wybrać do zestawienia interesujące nas wymiary, korzystając w tym celu z operacji *eTraverse*. Przy czym przez wymiar *MEASURE* rozumiemy tutaj zestawienie, otrzymane wyniku poprzedniej operacji *drill-down*. Zapis w postaci wyrażeń algebry relacji przedstawia się następująco:

$$eCascade\ drill-down(MEASURE, METER, \{\}, \{day\}) =$$
$$eTraverse(eDecompose(MEASURE, METER, \{\}), \{day\}) =$$
$$\pi_{measure.value,meter.id,meter\_model.scope,meter.medium,day.day,day.month...}(eTraverse((MEASURE \bowtie$$
$$eTraverse(METER, \{\})), \{day\}) = \pi_{measure.value,day.day,day.month...}(MEASURE \bowtie$$
$$(\pi_{meter.id,meter\_model.scope,meter.medium}(METER \bowtie METER\_MODEL)) \bowtie$$
$$SHJ(\{H\_TIME\}, \{day\})$$

Na rys. 13 przedstawiono przebieg operacji *drill-down* z użyciem przykładowych tabel.



Rys. 13. Drugi poziom operacji *eCascaded drill-down*
Fig. 13. Second level of *eCascaded drill-down* operation

Równoważne zapytanie SQL:

```
SELECT m.value, mt.id, mt.medium, mtm.scope, d.id d.day, day.month...
FROM meter mt, meter_model mtm,measure m, day d
WHERE m.date_id = d.id AND m.meter_id = mt.id AND mt.model_id = mtm.id
```

Poziom III (najbardziej szczegółowy) – data, wartość pomiaru, identyfikator licznika, jego zakres i mierzone medium, identyfikator klienta oraz sposób płatności

Na poziomie tym rozszerzone zostaje poprzednie zestawienie o dodatkowe informacje, dotyczące klienta oraz sposobu jego płatności. Należy więc wykonać operację *drill-down*, którego parametrami będą poprzednio otrzymane zestawienie oraz wymiar *CLIENT*. Zapis tej operacji za pomocą wyrażeń algebry relacji wyraża się:

$$eCascade\ drill-down(cascade\ drill-down(MEASURE, METER, \{\}, \{day\}), CLIENT, \{\}, \{\})$$

W celu ograniczenia złożoności zapisu możemy oznaczyć:

$$eCascade\ drill-down(MEASURE, METER, \{\}, \{day\}) = ES^C \qquad (*)$$

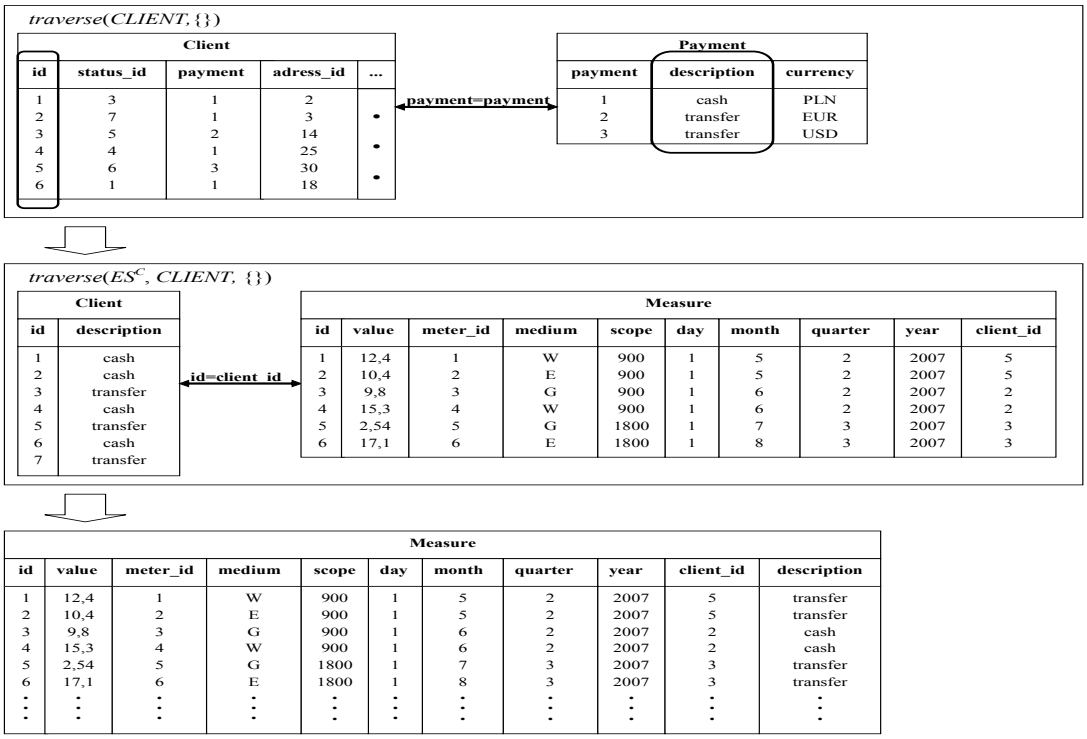
Po rozwinięciu operacji oraz uwzględnieniu podstawienia (*) mamy:

$$eTraverse(eDecompose(ES^C, CLIENT, \{\}), \{\}) =$$

$$\pi_{client.id,\,payment.descriotion}(eTraverse(ES^C \rhd\lhd\ eTraverse(CLIENT, \{\}), \{\}) =$$

$$\pi_{client.id,payment.descriotion}(ES^C \bowtie CLIENT \bowtie PAYMENT \bowtie SHJ(\{\},\{\}))$$

Na rys. 14 zaprezentowano krok po kroku wykonanie po raz trzeci operacji *eCascaded drill-down*. Wykorzystano wyniki poprzedniej operacji, aby nie zaciemniać rysunku.



Rys. 14. Trzeci poziom operacji *eCascaded drill-down*
Fig. 14. Third level of *eCascaded drill-down* operation

Równoważne zapytanie SQL:

```
SELECT m.value, d.id, mt.id, cl.client_id, p.description
FROM meter mt, measure m, day d, client cl, payment p
WHERE m.date_id = d.id AND m.meter_id = mt.id AND mt.client_id = cl.id AND
cl.payment = payment.payment
```

Przebieg operacji *eCascaded roll-up* odbywa się w odwrotnej kolejności.

2.8. eKaskadowe cięcie (eCascaded-slice) i eKaskadowe krojenie (eCascaded-dice)

Operacje te redukują liczbę wyświetlanych elementów dla kostki wielowymiarowej poprzez ustalenie dodatkowych warunków selekcji dla danych atrybutów podgwiazd (w przypadku operacji *eCascaded-slice* są to atrybuty pojedynczego wymiaru – gwiazdy lub płatka śniegu). Operacja *slice* wykonuje najpierw *eTraverse* na pojedynczej gwieździe, po czym przesuwa się o jeden poziom wyżej w celu wykonania dekompozycji. Operacja *dice,* w przeciwieństwie do *slice,* dokonuje selekcji na więcej niż jednej podgwieździe. Operacją dokonującą takiej selekcji jest skok, po wykonaniu którego następuje przesunięcie o poziom

dla celu wykonania dekompozycji. Poprzez kilkukrotne operacje skoku dokonuje się selekcji na więcej niż dwóch podgwiazdach. Zapis obu tych operacji w postaci algebraicznej to:

$$eCascaded\_slice(ES^{C}_{\{II-1\}}, S, I_S) = eDecompose(ES^{C}_{\{II-1\}}, S, I_S),$$

gdzie: S jest schematem S^S lub S^F znajdującym się na poziomie $M=k$, posiadającym zbiór wymiarów hierarchicznych H i będącym podgwiazdą $ES^{C}_{\{II-1\}}$ (M=k–1), a I_S jest zbiorem tabel, które dla danej hierarchii $H_S \in H$ określają graniczny poziom złączenia h_S w ramach tej hierarchii;

$$eCascaded\_dice(ES^{C}_{\{II-1\}}, S_1, S_2, I_1, I_2, I_{1,2}) = eDecompose(ES^{C}_{\{II-1\}}, eJump(S_1, S_2, I_1, I_2)), I_{1,2}),$$

gdzie: S_1 i S_2 są schematami S^S lub S^F znajdującymi się na poziomie $M=k$ i posiadającymi zbiór wymiarów hierarchicznych odpowiednio H_1 oraz H_2, są podgwiazdami $ES^{C}_{\{II-1\}}$ (M=k–1), a I_1 i I_2 są zbiorami tabel, które dla danych hierarchii $H_{S_1} \in H_1$ i $H_{S_2} \in H_2$ określają graniczny poziom złączenia h_{S_1} oraz h_{S_2} w ramach tej hierarchii, a I_1 i I_2 są zbiorami tabel określającymi graniczny poziom złączenia dla hierarchii należących do zbioru $H_1 \cup H_2$.

Poniżej znajduje się przykład wykonania operacji *eCascaded slice* dla tabeli *MEASURE* uzyskanej w poprzednim przykładzie dzięki wykonaniu operacji *eCascaded drill-down*. Z tabeli tej celowo usunięto informację o mierzonym medium, aby konieczna była operacja dekompozycji tabeli *MEASURE* poprzez tabelę *METER*.

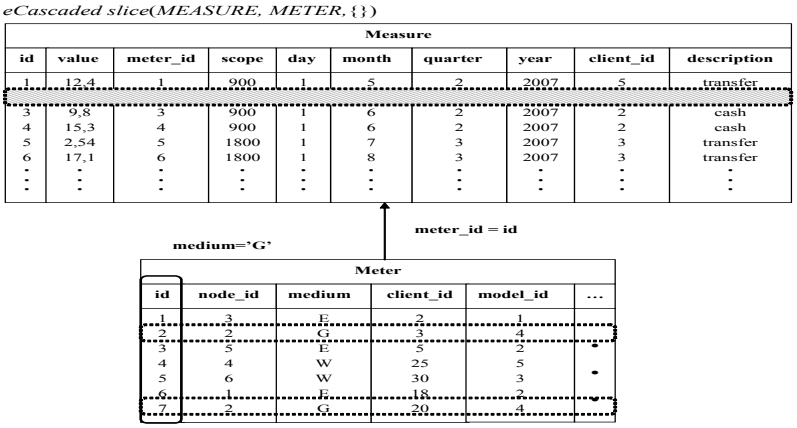
Zapytanie 5

Ograniczyć wyświetlane dane do liczników gazu.

Operację tę można zapisać następująco:

$$eCascade\ slice(MEASURE, METER, \{\}) = \pi_{measure.all}(\sigma_{meter.medium='G'}(MEASURE \rhd\lhd METER))$$

Rys. 15 prezentuje sposób wykonania operacji na przykładowych tabelach.



Rys. 15. Przykład wykonania operacji *eCascaded slice*
Fig. 15. Example of execution of *eCascaded slice* operation

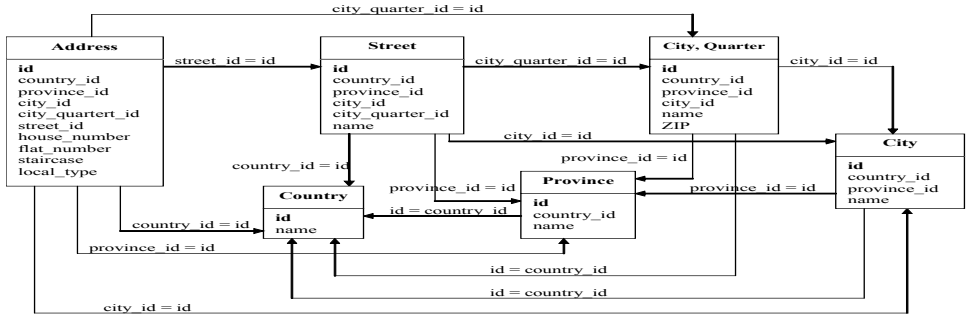
Jako ostatni przykład operacji COLAP omówiona zostanie operacja *cascaded dice* wykonana na zestawieniu przedstawionym na rys. 16. Zestawienie to zbudowane jest na

wymiarze kaskadowym *METER* oraz jego podwymiarach *CLIENT* oraz *NODE*. Wszystkie schematy wymienionych wymiarów zostały już wcześniej opisane.

Meter						
meter_id	**medium**	**scope**	**client_id**	**description**	**node_id**	**manufactured**
1	W	900	5	transfer	4	1
2	E	900	5	transfer	4	2
3	G	900	2	cash	3	3
4	W	900	**2**	cash	2	4
5	G	1800	3	transfer	3	5
6	E	1800	**3**	transfer	2	6
⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Rys. 16. Zestawienie wykonano bazując na wymiarach *METER*, *CLIENT* oraz *NODE*
Fig. 16. Comparison made basing on *METER*, *CLIENT* and *NODE* dimension

Na rys. 17 pokazano schemat wymiaru hierarchicznego *H_ADDRESS*, który jest jednym z podwymiarów wymiaru *CLIENT*. Został on tu przytoczony, ponieważ kolejny przykład będzie na nim oparty.



Rys. 17. Schemat wymiaru hierarchicznego *H_ADDRESS*
Fig. 17. Schema of hierarchical dimension *H_ADDRESS*

Zapytanie 6

Ograniczyć powyższe zestawienie tak, aby dotyczyło tylko klientów zamieszkałych w Gliwicach oraz węzłów, których data produkcji przypada na drugi kwartał 2007 roku.

Dla takiego zapytania ogranicza się liczbę wyświetlanych informacji z wykorzystaniem dwóch wymiarów, dlatego należy skorzystać z operacji *eCascaded dice*. Wykorzystywane są wymiary hierarchiczne *H_ADRESS* z wymiaru *CLIENT* oraz *H_TIME* z wymiaru *NODE*. Hierarchia adresu jest schematem płatka śniegu typu 2, w związku z czym trzeba wykonać złączenie tabel *ADDRESS* oraz *CITY*. Złączenie to jest wykonywane w ramach operacji *traverse*(*CLIENT,*{*city*}), która z kolei jest wykonywana jako jeden z kroków operacji *jump*(*NODE, CLIENT,* {*city*},{*day*}). Całą operację *eCascaded dice* można zapisać jako:

$$eCascade\ dice(METER, CLIENT, NODE, \{city\}, \{day\}, \{\}) =$$
$$\pi_{meter.all}(\sigma_{city.name='Gliwice', day.year=2007, day.quarter=2}(METER \rhd\lhd$$
$$eJump(CLIENT, NODE, \{city\}, \{day\}))$$

Po rozpisaniu operacji *eJump*:

$$\pi_{meter.all}(\sigma_{city...} METER \rhd\lhd eTraverse((eTraverse(CLIENT, \{city\}) \rhd\lhd$$
$$(eTraverse(NODE, \{day\}), \{\}))$$

Następnie po rozpisaniu operacji *eTraverse*:

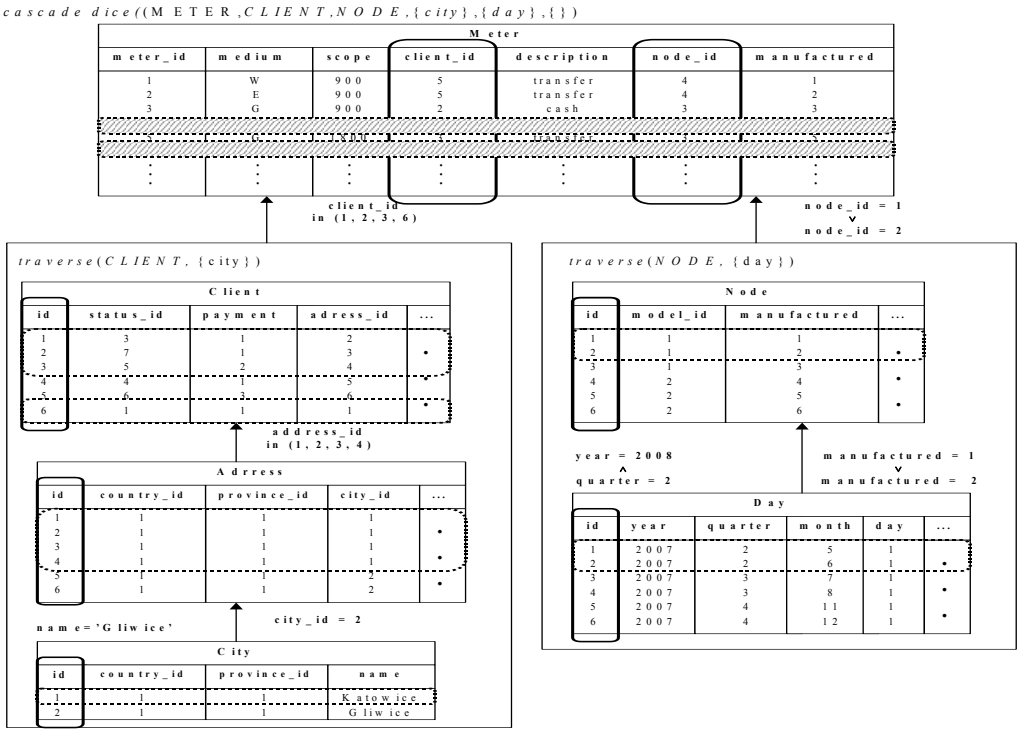
$$\pi_{meter.all}(\sigma_{city...}METER \bowtie eTraverse((CLIENT, SHJ(H\_ADDRESS, city) \bowtie (NODE, SHJ(H\_TIME, day), \{\})))$$

Rozwijając operację *SHJ*:

$$\pi_{...}(\sigma_{...}(METER \bowtie eTraverse(ADDRESS \bowtie CITY \bowtie CLIENT \bowtie METEl \bowtie NODE \bowtie DAY, \{\}))$$

Ostateczny wynik:

$$\pi_{...}(\sigma_{...}(METER \bowtie ADDRESS \bowtie CITY \bowtie CLIENT \bowtie METER \bowtie NODE \bowtie DAY, \{\}))$$



Rys. 18. Przykład wykonania operacji *eCascaded dice*
Fig. 18. Example of execution of *eCascaded dice* operation

2.9. MCUBE

MCUBE (ang. *multiple cubes*) wykonuje wielokrotne obliczenia typu CUBE na takich relacjach, jak tablice bazowe lub zmaterializowane widoki. W algebrze relacji operacja typu MCUBE wyraża się za pomocą wzoru:

$$MCUBE(Q) = op(V, eCascaded\_roll\_up(Q, P, I_q, I_p)),$$

gdzie: *op* jest relacyjnym operatorem agregacji takim jak *sum, V* jest zbiorem centralnych pomiarów *Q*, *P* jest zbiorem podgwiazd dla *Q*, którymi mogą być zarówno $ES^{C}_{\{H-1\}}$, S^{S}, jak i S^{SF} posiadające zbiory wymiarów hierarchicznych *H*, natomiast I_q i I_q są zbiorami tablic, które dla

danych zbiorów hierarchii $H_q \in H$ oraz $H_q \in H$, określają graniczny poziom złączenia h_p i h_q w ramach tych hierarchii.

Na podstawie powyższego wyrażenia można stwierdzić, iż *MCUBE* jest to operacja agregacji dla wielokrotnego kaskadowego zwijania.

3. Podsumowanie

Zdefiniowano podstawowe kaskadowe operacje ECOLAP dla schematu rozszerzonej gwiazdy kaskadowej $ES^{C}_{\{II-1\}}$. Przedstawione operacje ECOLAP typu: *drill-down*, *roll-up* oraz *slice & dice* oraz *MCUBE* pozwalają na analizę danych przy użyciu nowego modelu logicznego $ES^{C}_{\{II-1\}}$. Dzięki wykorzystaniu wyrażeń algebry relacji pokazano również, iż rozszerzony schemat gwiazdy kaskadowej może być zbudowany na tradycyjnych schematach gwiazdy oraz płatka śniegu.

Dla większości operacji przedstawiono przykłady, które omówiono zarówno z użyciem wyrażeń algebry relacji, jak również na tabelach SDW. W ten sposób umożliwiono prześledzenie krok po kroku działania danej operacji. Przykłady te dowodzą poprawności przedstawionych definicji oraz umożliwiają ich zrozumienie i weryfikację.

Dalsze prace będą skoncentrowane na zdefiniowaniu operacji ECOLAP dla innych schematów rozszerzonej gwiazdy kaskadowej oraz rozszerzeniu zbioru operacji o kolejne elementy, z uwzględnieniem funkcji eksploracji danych, tj.: klasyfikacji i predykcji.

LITERATURA

1. Gorawski M., Malczok M.: Materialized aR-tree in Distributed Spatial Data Warehouse. International Journal: Intelligent Data Analysis (IDA), ISSN: 1088-467X, IOS Press Vol.10, nr. 4, 2006, s. 361÷377.
2. Yu S., Atluri V., Adam N. R.: Cascaded star: A hyper-dimensional model for a data warehouse. 17th International Conference on Database and Expert Systems Applications, DEXA 2006, vol. 4080 LNCS, 2006, s. 439÷448.
3. Gorawski M.: Definiowanie schematów rozszerzonej gwiazdy kaskadowej. Konferencja Bazy Danych: Aplikacje i Systemy BDAS`07, 2007, s. 103÷114.
4. Gorawski M., Gębczyk M.: Distributed Approach of Continuous Queries with *k*nn Join Processing in Spatial Data Warehouse. 9th International Conference on Enterprise Information Systems (ICEIS 2007) l, Madeira – Portugal, 2007, s. 131÷136.

5. Gorawski M., Gorawski M.: Balanced Spatio-Temporal Data Warehouse with R-MVB, STCAT and BITMAP Indexes. PARELEC 2006 5-th International Symposium on Parallel Computing in Electrical Engineering, Poland, IEEE CS, 2006, s. 43÷48.
6. Gorawski M., Dyga M.: Indexing of Spatio-Temporal Telemetric Data based on Distributed Mobile Bucket Index. Parallel and Distributed Computing and Networks (PDCN 2006) as part of the 24th IASTED International Multi-Conference on APPLIED INFORMATICS, ACTA Press, 2006, s. 292÷297.

Recenzent: Dr hab. Zygmunt Mazur, prof. Pol. Wrocławskiej

Wpłynęło do Redakcji 14 sierpnia 2007 r.

Abstract

The paper proposes the definitions of *Expanded Cascaded OLAP* (ECOLAP) operations described with the relation algebra. The operations have been defined for the *Extended Cascaded Star* schema in spatial data warehouse SDW basing on the existing definitions of the COLAP operations. Moreover, they support extensions presented in a new logical model. In order of better understanding, every operation is illustrated with an example that shows, step by step, how the operation proceed. The examples also prove the correctness of presented definitions.

Adresy

Marcin GORAWSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, M.Gorawski@polsl.pl.
Marcin Bugdol: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, M.Bugdol@polsl.pl.