

Podstawowe operacje bitemporalne zapewniające zachowanie integralności bitemporalnej bazy danych z pełną historią zmian

1. Wstęp

Wiele systemów informatycznych realizowanych na potrzeby administracji publicznej, jak większość systemów wykorzystujących technologie baz danych, posiada temporalną naturę. Wynika ona z kilku zasadniczych faktów.

Po pierwsze, wartości cech (atrybutów) obiektów utrwalanych w bazach danych w ogólnym przypadku ulegają naturalnym zmianom w cyklu swojego życia. Po drugie, zmiany mogą ulegać powiązania pomiędzy obiektami utrwalonymi w bazie danych. Po trzecie, nie można wykluczyć wprowadzania błędnych danych, a co za tym idzie, konieczności ich korygowania. Wymóg pamiętania pełnej historii takich zmian w bazie danych wymaga zasadniczo odmiennego od klasycznego podejścia do projektowania bazy danych, jak i uwzględnienia aspektów temporalnych w aplikacji.

Część oferowanych obecnie komercyjnych systemów zarządzania bazą danych (takich firm, jak IBM², Oracle³, Microsoft⁴ czy Teradata⁵) dostarcza mechanizmów tworzenia relacji temporalnych oraz operacji temporalnych. Niemniej jednak każdy z nich posiada określone ograniczenia i niejednokrotnie nie będzie można wykorzystać dostarczanych właściwości wprost, aby spełnić

¹ Wojskowa Akademia Techniczna w Warszawie, Wydział Cybernetyki, Instytut Systemów Informatycznych.

² C. Saracco, M. Nicola, L. Gandhi, *A matter of time: Temporal data management in DB2 10*, <http://www.ibm.com/developerworks/data/library/techarticle/dm-1204db2temporaldata/> (dostęp: 20.04.2017).

³ L. Lance, T. Kyte, *Oracle Database Concepts, 11g Release 2 (11.2)*, 2015, http://docs.oracle.com/cd/E11882_01/server.112/e40540/title.htm (dostęp: 17.04.2017).

⁴ <http://pl.seequality.net/sql-server-2016-temporal-tables/> (dostęp: 19.06.2017).

⁵ R.T. Snodgrass, *A Case Study of Temporal Data*, Teradata Corporation, 2010, http://cs.ulb.ac.be/public/_media/teaching/infoh415/teradata_temporal_case_study.pdf (dostęp: 11.04.2017).

wymagania związane z zachowaniem pełnej historii zmian w konkretnym systemie informatycznym.

Celem pracy jest omówienie natury podstawowych operacji bitemporalnych (zapis, aktualizacja) zapewniających zachowanie integralności bitemporalnej relacyjnej bazy danych z pełną historią zmian. W przekonaniu autora, znajomość treści zawartych w pracy powinna ułatwić ocenę przydatności i dokonanie wyboru komercyjnego systemu zarządzania bazą danych do realizacji systemu informatycznego z pełną historią zmian. Jest to o tyle istotne, że aktualnie nie dopracowano się jeszcze ogólnie przyjętej metodyki projektowania temporalnych baz danych.

2. Bitemporalna baza danych z pełną historią zmian

Bitemporalna baza danych jest odmianą bazy relacyjnej⁶, w której część relacji to relacje bitemporalne. Relacja bitemporalna charakteryzuje się tym, że każda z jej krotek posiada znaczniki (stemple) czasowe określające czas, w jakim jej wartości są prawdziwe, odpowiednio, w świecie rzeczywistym oraz w bazie danych. Taka krotka reprezentuje pewien stan obiektu, czyli określa zbiór wartości jego atrybutów w czasie. Jeżeli znacznik czasowy określa czas, w którym obiekt znajduje się (lub się znajdował) w określonym stanie, w świecie rzeczywistym, nazywany jest znacznikiem czasu ważności (ang. *Valid Time*). Jeżeli znacznik czasowy określa czas, w którym obiekt znajduje się (lub się znajdował) w określonym stanie, w bazie danych, nazywany jest znacznikiem czasu transakcji (ang. *Transaction Time*). Każdy z tych znaczników zapisywany jest za pomocą jednego (znacznik zdarzeniowy) lub dwóch (znacznik przedziałowy) chrononów⁷ przyjmujących pewną wartość o zadanej granulacji czasu⁸. Należy zauważyć, że w relacji bitemporalnej identyfikator obiektu traci swą unikalność, ponieważ identyfikator ten występuje w każdym ze stanów obiektu. Unikalność klucza głównego relacji bitemporalnej uzyskuje się, dołączając do identyfikatora obiektu oba znaczniki czasowe⁹ lub przez wprowadzenie identyfikatora zastępczego (identyfikator

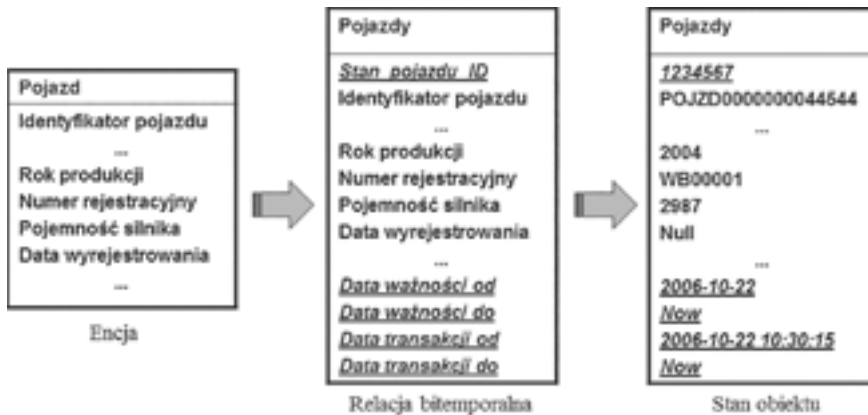
⁶ Taka definicja została przyjęta na potrzeby niniejszej pracy, w której ograniczono się tylko do relacyjnych baz danych.

⁷ Chronon – niepodzielna jednostka czasu.

⁸ Do dalszych rozważań przyjęto znaczniki przedziałowe, co nie zmniejsza ogólności rozważań.

⁹ T. Johnston, *Bitemporal Data. Theory and Practice*, Elsevier Inc., Waltham, USA 2014, s. 36.

pozycji), przyjmującego unikalną wartość dla każdego ze stanów obiektu¹⁰. Nie zmienia to faktu, że identyfikator obiektu nadal wskazuje obiekt, pełniąc funkcję swego spinacza stanów obiektu w relacji bitemporalnej.



Rysunek 1. Relacja bitemporalna obiektu *Pojazd* i aktualny stan tego obiektu (przykład)

Źródło: S. Rozmus, *Projektowanie baz danych z pełną historią zmian danych – model bitemporalnej bazy danych i operacje zapisu*, „Biuletyn WAT” 2016, t. 65, nr 1, s. 95.

Przykład relacji bitemporalnej dla obiektu reprezentowanego przez encję *Pojazd* oraz przykładowy stan tego obiektu przedstawia rysunek 1. Przykładowy stan obiektu, zapisany w tej relacji, jest stanem aktualnie obowiązującym. Fakt ten określa się przez nadanie symbolicznej wartości *Now* końcom przedziałów czasów obowiązywania, odpowiednio ważności i transakcji. Implementacja wartości *Now* zależy od przyjętego typu danych, przy czym muszą zostać spełnione następujące warunki:

- dla czasu ważności – zaimplementowana wartość nie zostanie osiągnięta podczas całego życia obiektu;
- dla czasu transakcji – zaimplementowana wartość nie zostanie osiągnięta podczas okresu eksploatacji bazy danych.

Zakładając, że w bazie danych konieczne będzie pamiętanie nie tylko kolejnych stanów obiektów wynikających z naturalnych zmian wartości ich atrybutów, ale również wszelkich korekt już zapisanych wartości atrybutów, bitemporalną bazą danych z pełną historią zmian nazwiemy bazą danych, w której wszystkie relacje są relacjami bitemporalnymi.

¹⁰ Sposób realizacji klucza głównego relacji bitemporalnej jest nieistotny z punktu widzenia dalszych rozważań.

3. Integralność bitemporalnej bazy danych z pełną historią zmian

Integralność bazy danych stanowi o tym, czy dany stan bazy jest dopuszczalny (poprawny) czy też nie. Szeroko rozumiana integralność danych obejmuje logiczną spójność danych (spójność wewnętrzna) oraz zgodność ze stanem świata rzeczywistego opisywanego przez te dane (spójność zewnętrzna). Integralność jest związana ze zmianami, jakie zachodzą w bazie danych. Zmiany mogą być spowodowane zarówno zdarzeniami zewnętrznymi (aplikacja użytkownika), jak i wewnętrznymi (operacje serwera bazy danych, zwane transakcjami). Zachowanie poprawności danych i związków między nimi realizuje się przez wymuszanie zdefiniowanych więzów integralności oraz stosowanie wyzwalaczy (ang. *triggers*). Więzy integralności to warunki, których spełnienie świadczy, że baza danych jest w stanie spójnym.

Więzy integralności dzielimy na:

- integralność encji, co oznacza, że każda relacja powinna mieć klucz główny o unikalnych wartościach;
- integralność krotki, co oznacza, że każda krotka opisuje jeden element świata rzeczywistego, przy czym na wartości krotki można nałożyć ograniczenia, które muszą być spełnione przez wszystkie krotki;
- integralność referencyjną, określającą powiązania pomiędzy obiektami, która oznacza, że klucz obcy może mieć wartość klucza głównego innej relacji w bazie (oznaczającą powiązanie pomiędzy obiektami), bądź wartość NULL, oznaczającą brak powiązania.

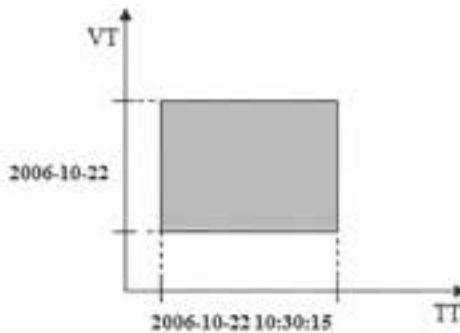
Tak zdefiniowane więzy integralności¹¹ nie znajdują zastosowania wprost w odniesieniu do bitemporalnych baz danych. Wynika to z faktu, że dane o poszczególnych obiektach zostają „z wielokrotnione” zgodnie z wymaganiami zachowania historii zmian dotyczących tych obiektów. Występuje zatem potrzeba wprowadzenia nowych definicji więzów integralności w odniesieniu do bitemporalnych baz danych z pełną historią zmian¹². Ze względu na ograniczenia redakcyjne co do objętości pracy dalsza jej część sprowadza się głównie do rozważań dotyczących

¹¹ Tak zdefiniowane więzy integralności odnoszą się do nietemporalnych baz danych.

¹² W przypadku bitemporalnych baz danych bez historii, tzn. takich, w których przechowywane są tylko stany aktualne, obowiązują więzy integralności zdefiniowane dla nietemporalnych baz danych.

zasad zapewnienia bitemporalnej integralności encji. Do tego celu wykorzystano koncepcję opartą o tzw. komórki bitemporalne (ang. *bitemporal cells*)¹³.

Komórka bitemporalna jest definiowana jako najmniejszy z możliwych do wyznaczenia obszar w układzie współrzędnych VT (oś czasu ważności), TT (oś czasu transakcji). Oznacza to, że komórka bitemporalna jest prostokątem, w którym długość jednego boku równa jest chrononowi czasu ważności, a drugiego – chrononowi czasu transakcji. Przykładowo, jeżeli przyjmiemy, że informacje o zmianach stanu obiektu będą otrzymywane z dokładnością do jednego dnia, natomiast zapisy zmian wprowadzane będą z dokładnością do jednej sekundy, komórka bitemporalna będzie prostokątem o bokach, odpowiednio, 1 dzień i 1 sekunda (rysunek 2).



Rysunek 2. Komórka bitemporalna (przykład)

Źródło: opracowanie własne.

Jak już wspomniano, czas trwania stanu obiektu w bitemporalnej bazie danych określa para znaczników: znacznik czasu ważności oraz znacznik czasu transakcji. W przypadku przyjętych znaczników przedziałowych można je opisać jako przedziały $\langle vt_p, vt_k \rangle$ (znacznik czasu ważności) oraz $\langle tt_p, tt_k \rangle$ (znacznik czasu transakcji),

gdzie:

vt_p oznacza początek okresu ważności stanu obiektu,

vt_k oznacza koniec okresu ważności stanu obiektu,

tt_p oznacza początek okresu transakcji stanu obiektu,

tt_k oznacza koniec okresu transakcji stanu obiektu.

Obszar wyznaczony przez pary $\langle vt_p, vt_k \rangle$, $\langle tt_p, tt_k \rangle$ krotki relacji bitemporalnej „pokryty” jest przez przylegające do siebie komórki bitemporalne, mieszczące

¹³ T. Johnston, op. cit., s. 117–126.

się w podanych przez te pary zakresach. Biorąc powyższe pod uwagę, bitemporalną integralność encji można zdefiniować następująco: bitemporalna integralność encji oznacza, że w relacji bitemporalnej nie występują krotki o tym samym identyfikatorze obiektu, które zajmowałyby tę samą komórkę bitemporalną, z wyjątkiem komórek powstających w wyniku zmiany stanu obiektu¹⁴.

Tak zdefiniowaną integralność encji można zapewnić przestrzegając określonych reguł w bitemporalnych operacjach zapisu, aktualizacji i usuwania krotek z relacji bitemporalnych. W przypadku baz danych oferujących możliwość definiowania relacji bitemporalnych, reguły te są implementowane w systemie zarządzania bazą danych. W przeciwnym przypadku reguły te można implementować w formie wyzwalaczy związanych z relacjami nietemporalnymi rozszerzonymi o znaczniki czasowe.

Istota bitemporalnych operacji zapisu i aktualizacji, których wykonanie zachowuje bitemporalną integralność encji została przedstawiona w dalszej części pracy. Omówienie bitemporalnych operacji usuwania krotek z relacji bitemporalnej wykracza poza ramy tej pracy.

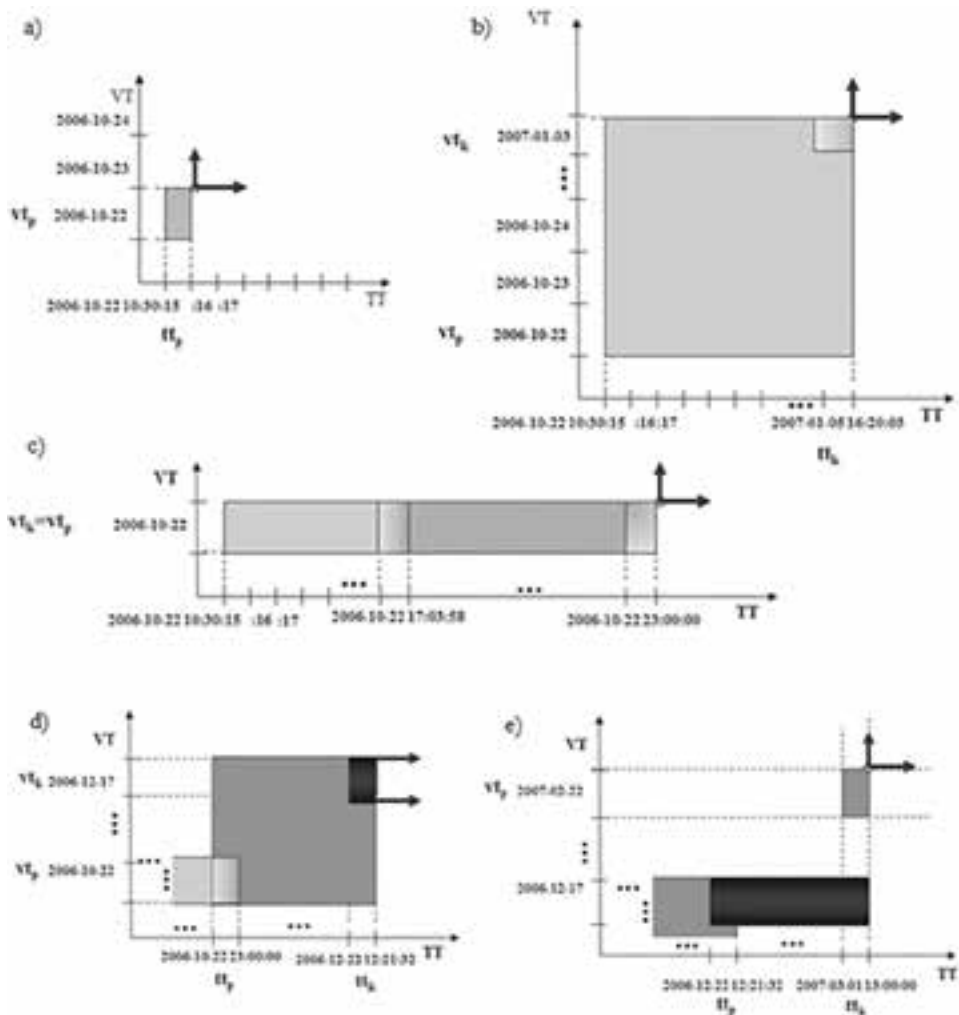
4. Bitemporalna operacja zapisu

Dane o obiekcie zostają wprowadzone do bazy danych po raz pierwszy, gdy zostanie wykonana pierwsza operacja zapisu, tworząc pierwszy stan tego obiektu (rysunek 3a). Początek okresu ważności obiektu określa chronon vt_p , początek czasu transakcji określa chronon tt_p . Oznacza to, że w momencie zapisu do bazy danych stan obiektu będzie trwał przez okres odpowiadający co najmniej jednemu chrononowi dla każdego z czasów. Stan ten staje się stanem aktualnym obiektu i trwa do chwili zmiany wartości co najmniej jednego z atrybutów obiektu (zmiana stanu) zakończenia życia obiektu lub przywrócenia obiektu do życia.

Z chwilą zajścia zmiany pojawia się nowy stan obiektu, który staje się stanem aktualnym, podczas gdy stan poprzedni staje się w tym samym momencie stanem historycznym (rysunek 3b). Widoczne strzałki oznaczają upływ czasu. Chronon, w którym nastąpiła zmiana ważności stanu obiektu, stanowi

¹⁴ Dodanie tego wyjątku, często pomijanego w literaturze, pozwala na zaakceptowanie faktu, że w momencie zmiany stanu obiektu, w bazie bitemporalnej mogą istnieć dwa różne stany (poprzedni i aktualny), co wynika jedynie z przyjętej dokładności dla znaczników czasowych.

ograniczenie górne vt_k przedziału ważności stanu poprzedniego, ale również ograniczenie dolne przedziału ważności stanu aktualnego. Podobnie chronon, w którym nastąpił zapis do bazy danych, stanowi ograniczenie górne tt_k przedziału transakcji stanu poprzedniego, ale również ograniczenie dolne przedziału transakcji stanu aktualnego. Należy tutaj wyraźnie podkreślić, że w przypadku zmian stanu obiektu w tym samym chrononie czasu ważności, w ramach tego chrononu wystąpi więcej niż jeden stan obiektu (rysunek 3c).



Rysunek 3. Graficzna ilustracja właściwości bitemporalnej operacji zapisu

Źródło: opracowanie własne.

W przypadku zakończenia życia obiektu (rysunek 3d) utworzenie nowego stanu jest wymuszane zdarzeniem, które powoduje, że nie przewiduje się dalszych, naturalnych zmian stanów tego obiektu w przyszłości (np. może to być decyzja administracyjna o wyrejestrowaniu pojazdu z powodu sprzedaży pojazdu za granicę). W nowo utworzonym stanie obiektu oba krańce przedziału ważności przyjmą tę samą wartość vt_k , natomiast tt_k przyjmie wartości *Now*. Obiekt (tzn. wszystkie dotychczas zapisane stany tego obiektu) pozostaje nadal w bazie danych, zaś upływ czasu dotyczy tylko czasu transakcji dla stanu, w którym obiekt zakończył życie.

W ogólnym przypadku obiekt może zostać przywrócony do życia (np. ponowne zarejestrowanie pojazdu zakupionego za granicą, który został wcześniej sprzedany za granicę). Widoczna przerwa w życiu obiektu (rysunek 3e) oznacza, że w tym czasie nie było wiadomo, co działo się z obiektem w rzeczywistości.

5. Bitemporalne operacje aktualizacji

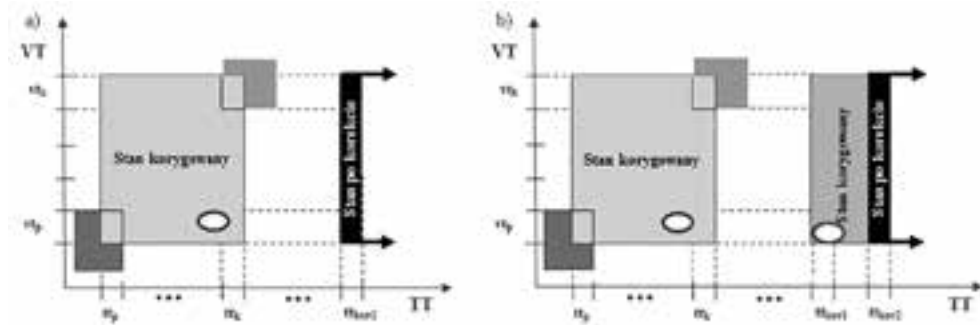
Bitemporalne operacje aktualizacji można podzielić na operacje korygowania wartości atrybutów wskazanego stanu oraz uzupełniania stanów historycznych.

Bitemporalna korekta stanu obiektu oznacza zmianę wartości jednego lub większej liczby atrybutów w tym stanie. W wyniku korekty stanu obiektu (stan korygowany) następuje utworzenie nowego stanu obiektu ze zmienionymi wartościami korygowanych atrybutów (stan po korekcie) oraz oznaczenie stanu korygowanego jako skorygowany (rysunek 4a)¹⁵. Wartości znaczników czasowych w korygowanym stanie nie ulegają zmianie. Znacznik czasu ważności w nowym stanie przyjmuje wartości znacznika czasu ważności ze stanu korygowanego $\langle vt_p, vt_k \rangle$, natomiast znacznik czasu transakcji przyjmuje wartości $\langle tt_{kor1}, Now \rangle$, gdzie tt_{kor1} oznacza czas dokonania korekty w bazie danych. Należy zauważyć, że tak utworzony stan nie jest stanem aktualnym obiektu. W przypadku kolejnej korekty znacznik czasu transakcji w tym stanie zmienia wartość *Now* na tt_{kor2} (czas dokonania kolejnej korekty w bazie danych), natomiast znacznik czasu transakcji w nowo utworzonym stanie przyjmuje wartości $\langle tt_{kor2}, Now \rangle$ (rysunek 4b).

W ogólnym przypadku bitemporalna korekta stanu obiektu może wymagać również dokonania korekty stanów poprzedzających korygowany stan, jak

¹⁵ Stan skorygowany oznaczono symbolicznie elipsą.

i stanów po nim następujących. Przed omówieniem tego zagadnienia, niezbędne jest wprowadzenie takich pojęć, jak stany równoważne (ang. *equivalent states*) i scalanie (ang. *coalescing*) stanów.



Rysunek 4. Graficzna ilustracja bitemporalnej korekty stanu obiektu

Źródło: opracowanie własne.

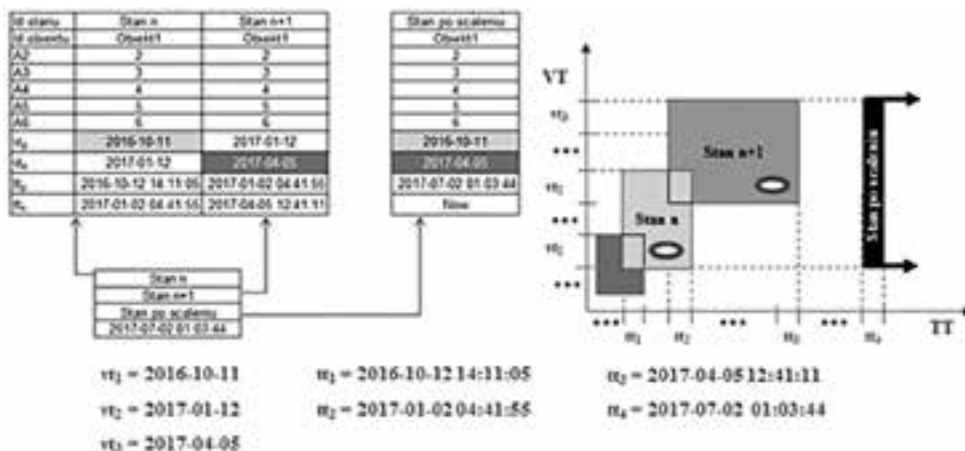
Dwa stany obiektu są równoważne, jeżeli wartości wszystkich odpowiadających sobie atrybutów obiektu są identyczne w obu tych stanach¹⁶. Wprowadzenie definicja bitemporalnej integralności encji nie ogranicza możliwości następowania po sobie stanów równoważnych, niemniej jednak ze względów wydajnościowych następujące po sobie stany równoważne powinny być łączone w jeden stan. Scalanie takich stanów równoważnych polega na:

- utworzeniu nowego stanu o wartościach atrybutów identycznych, jak w stanach równoważnych, i znacznikach czasowych wyznaczających przedział *<początek stanu poprzednika, koniec stanu następnika>* dla znacznika czasu ważności i przedział *<czas dokonania scalenia, Now>* dla czasu transakcji;
- wykonaniu zapisu informacji, z których stanów powstał scalony stan i w jakim czasie¹⁷;
- oznaczeniu stanów, które zostały scalone, jako skorygowane.

Przykład scalania następujących po sobie stanów równoważnych ilustruje rysunek 5. Występuje tu analogia do korekty pojedynczego stanu, z tym że znacznik czasu ważności stanu po korekcie określa przedział łącznego czasu trwania stanu obu scalanych stanów (tutaj $\langle vt_1, vt_3 \rangle$).

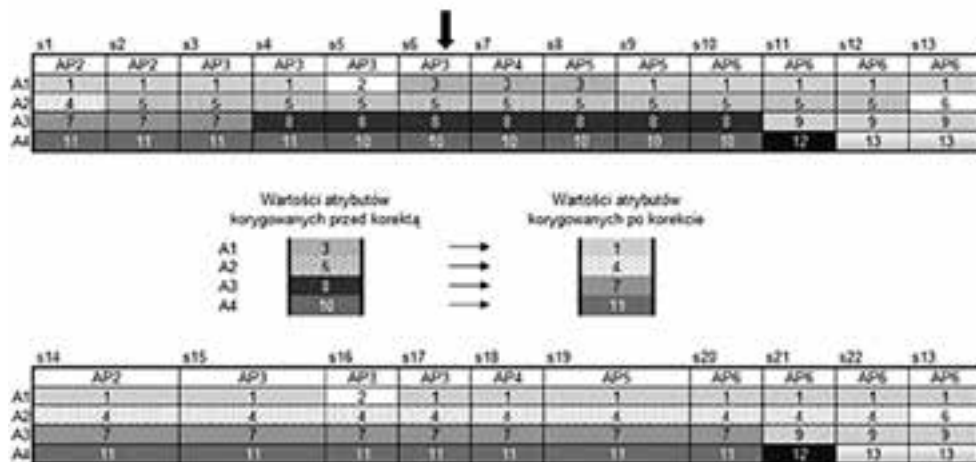
¹⁶ Identyfikator zastępczy (o ile występuje) oraz znaczniki czasowe z założenia nie są atrybutami obiektu.

¹⁷ Sposób zapisu nie jest istotny z punktu widzenia dalszych rozważań i nie będzie omawiany.



Rysunek 5. Scalanie stanów równoważnych obiektu (przykład)

Źródło: opracowanie własne.



Rysunek 6. Graficzna ilustracja wpływu korekty wartości atrybutów wskazanego stanu obiektu na stany sąsiednie

Źródło: opracowanie własne.

Wpływ bitemporalnej korekty pojedynczego stanu zostanie omówiony na przykładzie, w którym wykorzystano jedynie istotne elementy wymagane do wyjaśnienia tego zagadnienia (rysunek 6). Załóżmy, że w bazie danych została zapisana (przy użyciu wcześniej opisanej operacji zapisu) pewna sekwencja stanów s1 – s13. Należy dokonać korekty atrybutów A1 – A4 w stanie s6. Wartości

pozostałych atrybutów pozostają bez zmian, a ich wartości oznaczmy jako grupy AP_i ($i=1..6$), przy czym żadna z dowolnie wybranych par grup nie stanowi grup równoważnych w sensie równoważności stanów.

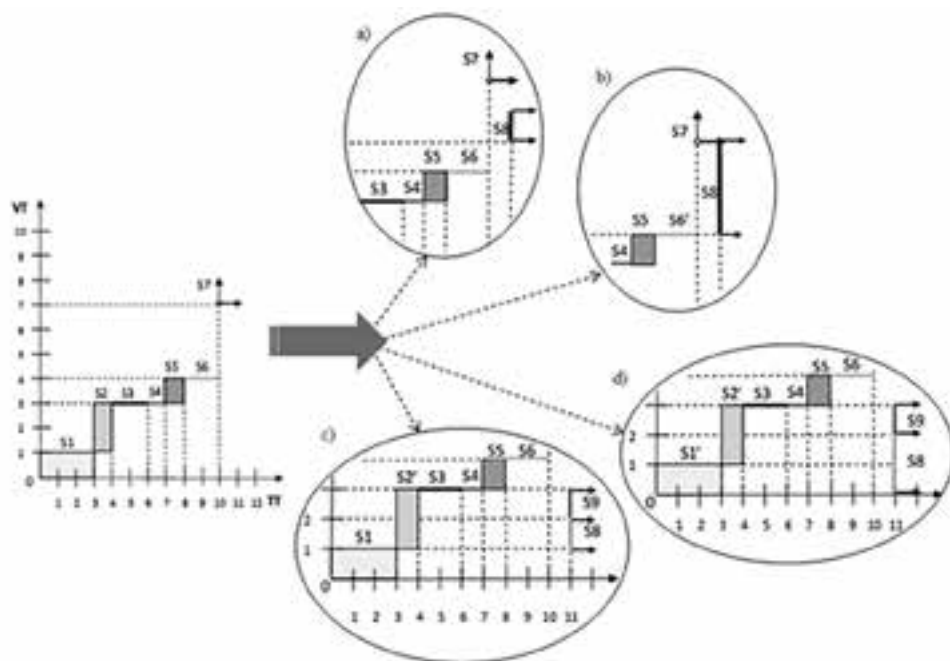
Można zauważyć, że korygowany stan s_6 pojawił się w następstwie zmiany wartości atrybutu A_1 , z wartości 2 na wartość 3. Wartości pozostałych atrybutów pozostały bez zmian. Zatem korygując wartości atrybutów w danym stanie, w stanie poprzedzającym, należy dokonać korekty tych wartości odpowiadających im atrybutów, które nie wymusiły zmiany stanu. Reguła ta odnosi do wszystkich stanów poprzedzających oraz następujących po stanie korygowanym. Jeżeli zatem po korekcie stanu s_6 wartość atrybutu A_1 w tym stanie przyjmie wartość 3, to dla stanów s_7 i s_8 również powinna zostać skorygowana na wartość 3. Korekta wartości atrybutu A_2 na 4 będzie skutkowałą korektą wartości tego atrybutu w stanach $s_2 - s_5$ i $s_7 - s_{12}$. Podobne korekty zostaną wykonane w odniesieniu do korygowanej wartości atrybutów A_3 i A_4 . W wyniku tak przeprowadzonej korekty zostaną utworzone nowe stany ($s_{14} - s_{22}$) uwzględniające scalenie stanów równoważnych. Należy zauważyć, że skorygowany stan s_2 i stan s_1 , są stanami równoważnymi. Po ich scaleniu stan s_1 staje się stanem skorygowanym, pomimo tego, że żadna z wartości jego atrybutów nie została skorygowana. W efekcie, po korekcie w bazie danych będą zapisane skorygowane stany $s_1 - s_{12}$, niezmieniony stan s_{13} oraz nowe stany $s_{14} - s_{22}$.

Uzupełnianie stanów historycznych jest operacją odmienną od korekty wartości atrybutów w danym stanie obiektu. Operacja ta polega na dopisaniu nowego stanu historycznego do już zapisanych stanów historycznych obiektu (rysunek 7).

Operacja uzupełnienia stanów historycznych obiektu może:

- nie zmieniać już zapisanych stanów historycznych obiektu, jeżeli okres ważności nowego obiektu zawiera się (rysunek 7a) lub jest równy (rysunek 7b) okresowi od zakończenia życia obiektu do ponownego powołania obiektu do życia;
- spowodować rozdzielenie (ang. *decoalescing*)¹⁸ już zapisanego stanu obiektu (rysunek 7c), ale również i scalenie stanu uzyskanego w wyniku rozdzielenia z już zapisanym stanem historycznym (rysunek 7d).

¹⁸ T. Johnston, op. cit., s. 166–167.



Rysunek 7. Graficzna ilustracja właściwości bitemporalnej operacji uzupełniania stanów historycznych

Źródło: opracowanie własne.

6. Podsumowanie i kierunki dalszych badań

W pracy omówiono naturę podstawowych operacji bitemporalnych (zapis, aktualizacja) zapewniających zachowanie integralności bitemporalnej relacyjnej bazy danych z pełną historią zmian. Przyjęta forma opisu ukierunkowana została na ułatwienie zrozumienia własności takich operacji i wskazanie niuansów, które trudno wychwycić studiując literaturę przedmiotu. Ze względu na ograniczoną dopuszczalną objętość pracy nie zostały w niej ujęte opisy pozostałych więzów integralności oraz inne operacje bitemporalne, na przykład anulowania zapisu czy też usuwania stanów obiektu. Są to kierunki dalszych badań, zmierzające do wypracowania spójnej, łatwej do zrozumienia i wykorzystania metodyki projektowania bitemporalnych baz danych.

Praca opiera się między innymi na doświadczeniach autora zdobytych przy budowie jednej z baz danych systemu CEPiK¹⁹.

Bibliografia

Johnston T., *Bitemporal Data. Theory and Practice*, Elsevier Inc., Waltham, USA 2014.
Rozmus S., *Projektowanie baz danych z pełną historią zmian danych – model bitemporalnej bazy danych i operacje zapisu*, „Biuletyn WAT” 2016, t. 65, nr 1, s. 89–109.

Źródła sieciowe

<http://pl.seequality.net/sql-server-2016-temporal-tables/> (dostęp: 19.06.2017).
Lance L., Kyte T., *Oracle Database Concepts, 11g Release 2 (11.2)*, 2015, http://docs.oracle.com/cd/E11882_01/server.112/e40540/title.htm (dostęp: 17.04.2017).
Saracco C., Nicola M., Gandhi L., *A matter of time: Temporal data management in DB2 10*, <http://www.ibm.com/developerworks/data/library/techarticle/dm-1204db2temporaldata/> (dostęp: 20.04.2017).
Snodgrass R.T., *A Case Study of Temporal Data*, Teradata Corporation, 2010, http://cs.ulb.ac.be/public/_media/teaching/infoh415/teradata_temporal_case_study.pdf (dostęp: 11.04.2017).

* * *

Basic Operations of a Bitemporal Database Ensuring the Integrity of the Database with a Full History of Changes

Summary

Maintaining the integrity of bitemporal databases is a very complex task. Database management systems have different support capabilities, but they do not always meet the needs of databases with a full history of changes. The purpose of this paper is to present the nature of basic bitemporal operations ensuring the integrity of a bitemporal database with a full history of changes.

Keywords: bitemporal relation, state of the object, valid time-stamp, transaction time-stamp.

¹⁹ Centralna Ewidencja Pojazdów i Kierowców.

