

Przetwarzanie danych temporalnych dla wymiaru czasu transakcyjnego na platformie MariaDB

Streszczenie. Celem artykułu jest wskazanie zakresu implementacji obsługi danych temporalnych dla czasu transakcyjnego w środowisku MariaDB oraz określenie stopnia zgodności tej implementacji z zapisami dotyczącymi temporalnych rozszerzeń języka SQL zawartych w standardzie ISO/IEC 9075 w wersji SQL:2011, a także prezentacja możliwości przetwarzania danych temporalnych w środowisku MariaDB.

Abstract. The aim of the article is an indication the scope of the implementation of temporal data support for transaction time in the MariaDB environment and determining the degree of compliance of this implementation with the provisions on temporal extensions of the SQL language of the ISO/IEC 9075 standard in the SQL: 2011 version, as well as to present possibility of processing temporal data by MariaDB environment. (Processing of temporal data for the transaction time on the MariaDB platform).

Słowa kluczowe: temporalne bazy danych, temporalne tabele, temporalne operatory, czas transakcyjny, SQL:2011, MariaDB.

Keywords: temporal databases, temporal tables, temporal operators, transaction time, SQL:2011, MariaDB.

Wstęp

Artykuł ten stanowi kolejny cykl rozważań na temat obsługi danych temporalnych w systemach baz danych opartych o relacyjny model danych. Poprzednie artykuły poświęcone zostały modelowaniu danych temporalnych w relacyjnym modelu danych [1], rozwojowi języka SQL i standardu ISO/IEC 9075 ze szczególnym uwzględnieniem składni pozwalającej składować i przetwarzać dane w RDBMS [2], a także możliwościom składowania i przetwarzania danych temporalnych na platformach MS SQL Server i Azure SQL Database [3], Oracle [4,5] oraz składowaniu danych temporalnych w wymiarze czasu transakcyjnego na platformie MariaDB [6]. Artykuł ten jest kontynuacją rozważań nt. możliwości składowania i przetwarzania danych temporalnych oferowanych przez serwery SQL. W całości poświęcony został on przetwarzaniu danych temporalnych na platformie MariaDB z uwzględnieniem wymiaru czasu transakcyjnego. Przeprowadzona została analiza zakresu możliwości przetwarzania danych temporalnych dostępnych na platformie MariaDB, dla czasu transakcyjnego. Celem tej analizy było zidentyfikowanie stopnia zgodności obsługi danych temporalnych z wymaganiami standardu ISO/IEC 9075 w szczególności z wersją standardu SQL:2011. Ponadto zaprezentowane zostały możliwości i przykłady przetwarzania danych temporalnych dla wymiaru czasu transakcyjnego w środowisku MariaDB.

Podstawowe wymagania składowania i obsługi danych temporalnych na podstawie specyfikacji standardu SQL:2011

Główne elementy dotyczące obsługi danych temporalnych, które zostały wprowadzone w standardzie SQL 2011 to [7-9]:

- definicja okresu czasu,
- tabele temporalne wersjonowane aplikacyjnie lub systemowo,
- tabele bitemporalne (wersjonowane aplikacyjnie i systemowo),
- możliwość aktualizacji i usuwania rekordów z określonego przedziału czasowego,
- temporalne ograniczenie klucza podstawowego,
- temporalne ograniczenie integralności referencyjnej,
- nowe predykaty czasowe dla interwałów czasowych.

Składowanie danych temporalnych w MariaDB

Obsługa danych temporalnych w środowisku MariaDB została wprowadzona w wersji 10.3.4, w 2017 roku.

Dotyczyła ona wymiaru czasu transakcyjnego w odniesieniu do tabel wersjonowanych systemowo. Rok później została zaimplementowana obsługa wymiaru czasu rzeczywistego w wersji 10.4.3, dla tabel wersjonowanych aplikacyjnie. W tej wersji także dostępna stała się obsługa tabel bitemporalnych [10].

Tabele wersjonowane systemowo dla obsługi czasu transakcyjnego przechowują oprócz aktualnych danych także historię wszystkich wprowadzonych zmian. Pozwala to analizować dane dla dowolnej chwili czasowej oraz porównywać dane dla wybranych punktów w czasie. Ponadto mechanizm ten oferuje rozbudowaną składnię dla zapytań temporalnych, a także gwarantuje niezmiennosc danych historycznych. Silnik bazodanowy automatycznie generuje wartości daty i czasu bądź też numeru identyfikacyjnego transakcji (w zależności od konstrukcji tabeli temporalnej), dla każdej instrukcji DML modyfikującej dane, operującej na tabeli temporalnej. Użytkownik nie może w żaden sposób ingerować w te wartości.

Zestawienie wybranych cech temporalnych standardu SQL:2011, oraz tych zaimplementowanych na platformie MariaDB, w odniesieniu do obsługi danych temporalnych dla wymiaru czasu transakcyjnego przedstawiono w tabeli 1.

Tabela 1. Spełnienie wybranych wymogów standardu SQL:2011, dla obsługi danych temporalnych w wymiarze czasu transakcyjnego

Wymagania	SQL:2011	MariaDB 10.6.5
Definicja czasookresu	✓	✓
Tabele wersjonowane systemowo	✓	✓
Implementacja predykatów czasowych TWS ¹	✓	✓
Jawne znaczniki czasowe	✓	✓
Niejawne znaczniki czasowe	✓	✓
Typ danych data	DATE, TIME, TIMESTAMP, INTERVAL	DATE, TIME, TIMESTAMP

Platforma MariaDB oferuje praktycznie pełną zgodność obsługi danych temporalnych w świetle wymagań zdefiniowanych w standardzie SQL:2011, w odniesieniu do wymiaru czasu transakcyjnego. Możliwe jest tworzenie temporalnych tabel wersjonowanych systemowo z jawnymi i

¹ TWS-tabela wersjonowana systemowo

niejawnymi znacznikami czasu. Dostępna jest także pełna gama predykatów czasowych umożliwiających przetwarzanie danych temporalnych, składowanych w tabelach temporalnych wersjonowanych systemowo. Ponadto zbiór ten jest rozszerzony o dodatkowy predykat temporalny ALL, w stosunku do wymagań zawartych w specyfikacji standardu SQL:2011. Środowisko MariaDB udostępnia także trzy z czterech typów danych zdefiniowanych w standardzie SQL:2011, oprócz typu INTERVAL.

Manipulacja danymi temporalnymi

Instrukcja INSERT skutkuje dodaniem nowego rekordu do tabeli temporalnej, który jest jednocześnie rekordem bieżącym. Poniżej przedstawiono przykład dodania kilku rekordów do tabeli temporalnej z obsługą czasu transakcyjnego za pomocą instrukcji INSERT jedno i wielopoleceniowej:

```
INSERT INTO pracownik (idpracownika, imie, nazwisko, miejscowosc)
VALUES (1, 'Jan', 'Kowalski', 'Warszawa');
```

```
INSERT INTO pracownik (idpracownika, imie, nazwisko, miejscowosc)
VALUES
```

```
(2, 'Katarzyna', 'Nowakowska', 'Olkusz'),
(3, 'Adam', 'Grzegorzczak', 'Gdynia'),
(4, 'Alicja', 'Jarebska', 'Jelenia Góra'),
(5, 'Julia', 'Widawska', 'Kraków');
```

Na rysunku 1 przedstawiono zawartość tabeli temporalnej po dodaniu rekordów, z widoczną zawartością kolumn dla jawnych znaczników czasowych.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart	SysCzasKoniec
1	Jan	Kowalski	Warszawa	2022-02-24 19:37:43.648392	2038-01-19 04:14:07.999999
2	Katarzyna	Nowakowska	Olkusz	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
3	Adam	Grzegorzczak	Gdynia	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
4	Alicja	Jarebska	Jelenia Góra	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
5	Julia	Widawska	Kraków	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999

Rys.1. Przykładowa zawartość tabeli temporalnej wersjonowanej systemowo, po dodaniu rekordów, źródło: opracowanie własne

W kolumnie składującej początek przedziału czasowego przechowywana jest wartość daty i czasu na chwilę dodania rekordu do tabeli. W kolumnie składującej koniec przedziału czasowego przechowywana jest górna wartość graniczna daty i czasu dla systemowego typu danych TIMESTAMP. Wartość ta jednocześnie oznacza, że jest to aktualna wersja rekordu.

W identyczny sposób dodawane są rekordy do tabeli temporalnej pracownik_trn, zawierającej znaczniki w postaci identyfikatorów transakcji, zamiast stempla czasowego. Na rysunku 2 przedstawiono zawartość tabeli temporalnej ze znacznikami, w postaci identyfikatorów transakcji, bezpośrednio po dodaniu rekordów.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart_IdTransakcji	SysCzasKoniec_IdTransakcji
1	Jan	Kowalski	Warszawa	1 034	18 446 744 073 709 551 615
2	Katarzyna	Nowakowska	Olkusz	1 038	18 446 744 073 709 551 615
3	Adam	Grzegorzczak	Gdynia	1 043	18 446 744 073 709 551 615
4	Alicja	Jarebska	Jelenia Góra	1 043	18 446 744 073 709 551 615
5	Julia	Widawska	Kraków	1 043	18 446 744 073 709 551 615

Rys.2. Zawartość tabeli temporalnej wersjonowanej systemowo po dodaniu rekordów, z uwidocznionymi znacznikami transakcji, źródło: opracowanie własne

Kolumna SysCzasStart_IdTransakcji przechowuje wartość identyfikatora transakcji, w ramach, której dodano rekord lub grupę rekordów do tabeli temporalnej. W kolumnie SysCzasKoniec_IdTransakcji przechowywana jest maksymalna dostępna wartość dla identyfikatora transakcji. Jednocześnie oznacza ona, że jest to aktualna wersja tego rekordu.

Instrukcja UPDATE umożliwia zaktualizowanie bieżącej wersji rekordu do jego nowej wersji, która staje się nową, obowiązującą wersją bieżącą. W kolumnie SysCzasStart

dla nowo zmodyfikowanej wersji rekordu generowana jest wartość znacznika czasowego odpowiadająca chwili przeprowadzenia modyfikacji. W kolumnie SysCzasKoniec generowana jest maksymalna wartość dla systemowego typu TIMESTAMP. Poprzednia wersja zmodyfikowanego rekordu zostaje zapamiętana w tabeli temporalnej. Wartość znacznika końca okresu dla rekordu historycznego jest równa wartości początku okresu dla rekordu bieżącego. Konwencjonalna postać zapytania o dane bieżące, w swoich wynikach będzie uwzględniać zaktualizowaną, bieżącą wersję rekordu. Poprzednia wersja rekordu sprzed modyfikacji, będzie dostępna dla zapytań odwołujących się do danych temporalnych. Poniżej przedstawiono przykład aktualizacji rekordu tabeli temporalnej:

```
UPDATE pracownik SET miejscowosc='Sanok' WHERE miejscowosc='Warszawa'
```

Na rysunku 3 przedstawiono wersję historyczną sprzed aktualizacji rekordu i aktualną wersję rekordu po jego modyfikacji. Obie wersje rekordu składowane są w tabeli temporalnej. Ponadto uwidoczniono zawartość kolumn dla jawnych znaczników czasowych.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart	SysCzasKoniec
1	Jan	Kowalski	Warszawa	2022-02-24 19:37:43.648392	2022-02-25 19:09:39.611159
1	Jan	Kowalski	Sanok	2022-02-25 19:09:39.611159	2038-01-19 04:14:07.999999

Rys.3. Aktualna i temporalna postać rekordu po jego aktualizacji, źródło: opracowanie własne

W identyczny sposób możliwe jest zaktualizowanie rekordów składowanych w tabeli temporalnej pracownik_trn zawierającej znaczniki, w postaci identyfikatorów transakcji, zamiast stempla czasowego. Na rysunku 4 przedstawiono wersję historyczną rekordu sprzed jego aktualizacji oraz jego wersję bieżącą, po przeprowadzonej modyfikacji. Wersja aktualna zmodyfikowanego rekordu otrzymała nową wartość identyfikatora transakcji, która jest składowana w kolumnie SysCzasStart_IdTransakcji. Wartość ta jednocześnie określa zawartość kolumny SysCzasKoniec_IdTransakcji dla poprzedniej wersji tego rekordu.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart_IdTransakcji	SysCzasKoniec_IdTransakcji
1	Jan	Kowalski	Warszawa	1 034	1 048
1	Jan	Kowalski	Sanok	1 048	18 446 744 073 709 551 615

Rys.4. Aktualna i temporalna postać rekordu po jego aktualizacji, z widocznymi znacznikami składującymi identyfikatory transakcji, źródło: opracowanie własne

Instrukcja DELETE ma zastosowanie tylko do danych aktualnych. Nie usuwa trwale rekordu z tabeli temporalnej wersjonowanej systemowo. Usunięty rekord staje się niedostępny dla klasycznej postaci zapytania wybierającego. Będzie on jednak dostępny dla zapytań odwołujących się do danych temporalnych.

Wartość znacznika końca okresu modyfikowana jest na wartość, odpowiadającą chwili czasowej, do kiedy dany rekord istniał w bazie danych, czyli na moment jego usunięcia. Znacznik początku okresu nie ulega zmianie. Poniżej przedstawiono przykład usunięcia rekordu z tabeli temporalnej:

```
DELETE FROM pracownik WHERE miejscowosc='Sanok';
```

Na rysunku 5 przedstawiono wersje historyczne rekordu sprzed aktualizacji, po jego modyfikacji a następnie jego usunięciu. Wszystkie wersje rekordu składowane są w tabeli temporalnej. Ponadto uwidoczniono zawartość kolumn dla jawnych znaczników czasowych.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart	SysCzasKoniec
1	Jan	Kowalski	Warszawa	2022-02-24 19:37:43.648392	2022-02-25 19:09:39.611159
1	Jan	Kowalski	Sanok	2022-02-25 19:09:39.611159	2022-02-25 19:32:03.010730

Rys.5. Temporalne wersje rekordu sprzed i po jego aktualizacji oraz po jego usunięciu, źródło: opracowanie własne

W identyczny sposób możliwe jest usunięcie rekordu z tabeli temporalnej pracownik_trn zawierającej znaczniki, w postaci identyfikatorów transakcji, zamiast stempla czasowego. Na rysunku 6 przedstawiono historyczne wersje rekordu po jego modyfikacji i usunięciu. Kolumnie końca przedziału czasowego dla usuniętego rekordu została przypisana nowa wartość identyfikatora transakcji, w ramach, której dany rekord został usunięty.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart_IdTransakcji	SysCzasKoniec_IdTransakcji
1	Jan	Kowalski	Warszawa	1 034	1 048
1	Jan	Kowalski	Sanok	1 048	1 053

Rys.6. Temporalne wersje rekordu po jego aktualizacji i usunięciu, z widocznymi znacznikami składającymi identyfikatory transakcji, źródło: opracowanie własne

Obsługa zapytań

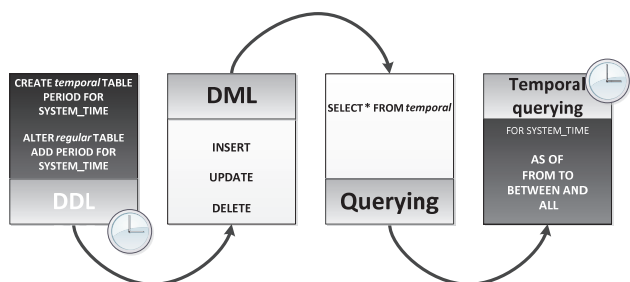
Na potrzeby zapytań temporalnych instrukcja wybierająca SELECT została rozbudowana o nowe predykaty temporalne. W zależności od zastosowanych predykatów temporalnych możliwe jest pobieranie rekordów tylko historycznych a więc aktualnych na dany moment w czasie, czy też do określonego punktu w czasie, bądź też rekordów aktualnych jak i historycznych. Wybór konkretnego predykatu temporalnego w zapytaniu będzie także determinował obszar przeszukiwania tabeli temporalnej, w przypadku jej podziału na partycje (czy przeszukiwana będzie tylko partycja historyczna, czy bieżąca i historyczna).

W środowisku MariaDB wprowadzone zostały klauzule do przetwarzania danych temporalnych składowanych w tabelach temporalnych wersjonowanych systemowo. Ich postać jest zgodna z zapisem standardu SQL:2011 (który wprowadza trzy klauzule) [2,8]. Na platformie MariaDB zaimplementowany został dodatkowy predykat temporalny. Łącznie dostępne są cztery konstrukcje umożliwiające temporalne odpytywanie tabel wersjonowanych systemowo (z obsługą czasu transakcyjnego) [11]. Zestawienie tych konstrukcji zawarte zostało w tabeli 2.

Tabela 2. Rozszerzenia składniowe do przetwarzania danych składowanych w tabelach temporalnych wersjonowanych systemowo na platformie MariaDB

SQL:2011	MariaDB-TT
AS OF	AS OF
FROM TO	FROM TO
BETWEEN AND	BETWEEN AND
-	ALL

Na rysunku 7 przedstawiony został sposób pracy z tabelami temporalnymi wersjonowanymi systemowo w środowisku MariaDB z obsługą czasu transakcyjnego.



Rys.7. Sposób pracy z tabelami temporalnymi wersjonowanymi systemowo w MariaDB, źródło: opracowanie własne

Zapytania na tabelach wersjonowanych systemowo z obsługą czasu transakcyjnego

Zapytania temporalne na tabelach z obsługą czasu transakcyjnego realizowane są za pomocą standardowej instrukcji SELECT z uwzględnieniem rozszerzenia FOR SYSTEM_TIME oraz predykatów temporalnych zestawionych w tabeli 2. Ich użycie pozwala na wybór

aktualnej jak i historycznej wersji rekordów. Zastosowanie instrukcji SELECT w jej podstawowej formie (bez rozszerzeń temporalnych) do tabeli temporalnej zwraca dane bieżące.

Poniżej przedstawiono przykład zastosowania klasycznej postaci instrukcji wybierającej do odpytania tabeli temporalnej, z jawnymi znacznikami czasu, bez użycia klauzuli FOR SYSTEM_TIME. Zwracane są tylko aktualne wersje rekordów.

```
SELECT * FROM pracownik;
```

Na rysunku 8 przedstawiono zestaw wyników dostępnych dla klasycznej postaci zapytania wybierającego po usunięciu rekordu. Usunięty rekord nie jest widoczny. Dostępne są tylko aktualne wartości.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart	SysCzasKoniec
2	Katarzyna	Nowakowska	Olkusz	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
3	Adam	Grzegorzczak	Gdynia	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
4	Alicja	Jarebska	Jelenia Góra	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
5	Julia	Widawska	Krakow	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999

Rys.8. Zestaw danych zwróconych przez zapytanie wybierające w klasycznej postaci. Usunięty rekord jest niewidoczny, źródło: opracowanie własne

Na rysunku 9 przedstawiono zestaw zwracanych wyników po aktualizacji rekordu (jeszcze przed usunięciem rekordu), z widoczną zawartością kolumn dla jawnych znaczników czasowych, dla konwencjonalnej postaci zapytania wybierającego. Dostępne są tylko wartości aktualne.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart	SysCzasKoniec
1	Jan	Kowalski	Sanok	2022-02-25 19:09:39.611159	2038-01-19 04:14:07.999999
2	Katarzyna	Nowakowska	Olkusz	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
3	Adam	Grzegorzczak	Gdynia	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
4	Alicja	Jarebska	Jelenia Góra	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
5	Julia	Widawska	Krakow	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999

Rys.9. Zestaw danych zwróconych przez zapytanie wybierające w klasycznej postaci, jeszcze przed usunięciem rekordu. Dane temporalne są niewidoczne, źródło: opracowanie własne

Poniżej przedstawiono przykład zastosowania klasycznej postaci instrukcji wybierającej do odpytania tabeli temporalnej ze znacznikami w postaci identyfikatorów transakcji, bez użycia klauzuli FOR SYSTEM_TIME. Zwracane są tylko aktualne wersje rekordów.

```
SELECT * FROM pracownik_trn;
```

Na rysunku 10 przedstawiono zestaw wyników dostępnych dla klasycznej postaci zapytania wybierającego po usunięciu rekordu, z widoczną zawartością kolumn przechowujących identyfikatory transakcji. Usunięty rekord nie jest widoczny. Dostępne są tylko aktualne wartości.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart_IdTransakcji	SysCzasKoniec_IdTransakcji
2	Katarzyna	Nowakowska	Olkusz	1 038	18 446 744 073 709 551 615
3	Adam	Grzegorzczak	Gdynia	1 043	18 446 744 073 709 551 615
4	Alicja	Jarebska	Jelenia Góra	1 043	18 446 744 073 709 551 615
5	Julia	Widawska	Krakow	1 043	18 446 744 073 709 551 615

Rys.10. Zestaw danych zwróconych przez zapytanie wybierające w klasycznej postaci, ze znacznikami w postaci identyfikatorów transakcji. Dane temporalne są niewidoczne, źródło: opracowanie własne

Na rysunkach 11 i 12 przedstawiono domyślną postać zwracanych wyników dla zapytania odwołującego się do tabeli temporalnej utworzonej z niejawnie definiowanymi znacznikami czasu. Domyślnie zawartość kolumn przechowujących znaczniki czasowe nie jest udostępniana. Zawartość tych kolumn staje się dostępna dopiero po uwzględnieniu ich nazw ROW_START i ROW_END na liście pól w instrukcji SELECT. W celu udostępnienia zawartości kolumn znaczników czasowych z tabeli temporalnej utworzonej, z jawnie definiowanymi kolumnami dla znaczników czasowych, ze statusem INVISIBLE, należy się posłużyć nazwami własnymi dla tych kolumn, które

zostały określone na etapie tworzenia tabeli temporalnej. W przypadku tabel temporalnych z jawnie zdefiniowanymi kolumnami znaczników czasowych, z domyślnym statusem ich widoczności, zawartość tych kolumn jest domyślnie dostępna dla temporalnego zapytania wybierającego, jak również dla zapytania w jego klasycznej postaci [12].

Poniżej przedstawiono przykład zwracający aktualne wersje rekordów z tabeli temporalnej z niejawnie zdefiniowanymi znacznikami czasu. Kolumny znaczników czasowych domyślnie są niewidoczne.

```
SELECT * FROM pracownik
```

IdPracownika	Imie	Nazwisko	Miejscowosc
1	Jan	Kowalski	Warszawa
2	Katarzyna	Nowakowska	Olkusz
3	Adam	Grzegorzczak	Gdynia
4	Alicja	Jarebska	Jelenia Góra
5	Julia	Widawska	Kraków

Rys.11. Domyślna postać zwracanych wyników z tabeli temporalnej z niejawnie definiowanymi znacznikami czasu, źródło: opracowanie własne

Poniżej przedstawiono przykład zwracający dodatkowo zawartość kolumn niejawnych znaczników czasowych:

```
SELECT *,ROW_START, ROW_END FROM pracownik;
```

IdPracownika	Imie	Nazwisko	Miejscowosc	ROW_START	ROW_END
1	Jan	Kowalski	Warszawa	2022-02-28 19:58:46.350073	2038-01-19 04:14:07.999999
2	Katarzyna	Nowakowska	Olkusz	2022-02-28 19:58:46.368851	2038-01-19 04:14:07.999999
3	Adam	Grzegorzczak	Gdynia	2022-02-28 19:58:46.368851	2038-01-19 04:14:07.999999
4	Alicja	Jarebska	Jelenia Góra	2022-02-28 19:58:46.368851	2038-01-19 04:14:07.999999
5	Julia	Widawska	Kraków	2022-02-28 19:58:46.368851	2038-01-19 04:14:07.999999

Rys.12. Postać zwracanych wyników z tabeli temporalnej z niejawnie definiowanymi znacznikami czasu, z jednoczesnym zwróceniem zawartości tych kolumn, źródło: opracowanie własne

Kolejny przykład umożliwia zwrócenie dodatkowo zawartości kolumn jawnych znaczników czasowych, których widoczność jest domyślnie wyłączona.

```
SELECT *,SysCzasStart, SysCzasKoniec pracownik;
```

Postać zwracanych wyników jest identyczna jak dla tabeli temporalnej z jawnie zdefiniowanymi znacznikami czasowymi, z domyślnym statusem widoczności tych kolumn.

Rozszerzenie temporalne AS OF pozwala uzyskać stan rekordów na dany moment w czasie, którego wartość jest podana, jako argument predykatu [2,8,11]. Poniżej przedstawiono przykład zapytania z użyciem klauzuli AS OF dla chwili czasowej sprzed wprowadzenia jakichkolwiek modyfikacji rekordów:

```
SELECT * FROM pracownik FOR SYSTEM_TIME AS OF  
TIMESTAMP'2022-02-25 18:30:30';
```

Na rysunku 13 przedstawiono zestaw wyników po aktualizacji rekordu, z widoczną zawartością kolumn dla jawnych znaczników czasowych, dla zapytania temporalnego, zwracającego stan rekordów dla zadanej chwili czasowej, sprzed jego aktualizacji. Dostępna jest tylko historyczna postać rekordu, która była aktualną jego postacią, w danej chwili czasowej oraz wersje bieżące rekordów, które nie zostały jeszcze zmodyfikowane. W zbiorze wynikowym niedostępna jest najnowsza (bieżąca) wersja rekordu. W kolumnie SysCzasKoniec widnieje wartość, do kiedy dana wersja rekordu była wersją obowiązującą.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart	SysCzasKoniec
1	Jan	Kowalski	Warszawa	2022-02-24 19:37:43.648392	2022-02-25 19:09:39.611159
2	Katarzyna	Nowakowska	Olkusz	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
3	Adam	Grzegorzczak	Gdynia	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
4	Alicja	Jarebska	Jelenia Góra	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
5	Julia	Widawska	Kraków	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999

Rys.13. Zestaw danych zwróconych przez temporalne zapytanie wybierające dla zadanej chwili czasowej. Widoczne są dane bieżące rekordów oraz poprzednia wersja rekordu sprzed jego aktualizacji (aktualna w zadanej chwili czasowej), źródło: opracowanie własne

Poniżej przedstawiono przykład zapytania z użyciem klauzuli AS OF dla chwili czasowej po wprowadzeniu modyfikacji rekordu:

```
SELECT * FROM pracownik FOR SYSTEM_TIME AS OF  
TIMESTAMP'2022-02-25 19:30:30';
```

Na rysunku 14 przedstawiono zestaw wyników zwróconych przez zapytanie temporalne, odzwierciedlający stan rekordów na zadany moment w czasie, tuż po wprowadzeniu modyfikacji. W odróżnieniu od rysunku 9, który odzwierciedla zawartość tabeli temporalnej sprzed usunięcia pierwszego rekordu (rekord nie był jeszcze fizycznie usunięty), w kolumnie SysCzasKoniec widnieje wartość, do kiedy dany rekord był rekordem aktualnym, w tym przypadku, kiedy został usunięty.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart	SysCzasKoniec
1	Jan	Kowalski	Sanek	2022-02-25 19:09:39.611159	2022-02-25 19:32:03.010730
2	Katarzyna	Nowakowska	Olkusz	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
3	Adam	Grzegorzczak	Gdynia	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
4	Alicja	Jarebska	Jelenia Góra	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
5	Julia	Widawska	Kraków	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999

Rys.14. Zestaw danych zwróconych przez temporalne zapytanie wybierające dla zadanej chwili czasowej. Widoczne są dane bieżące rekordów oraz wersja rekordu po jego aktualizacji, źródło: opracowanie własne

Poniższe zapytanie zwraca wersję rekordu dla transakcji o zadanej wartości identyfikatora [13]:

```
SELECT * FROM pracownik_trn FOR SYSTEM_TIME AS OF  
TRANSACTION 1034;
```

Na rysunku 15 przedstawiono rezultat działania tego zapytania. Zwracany jest jednoelementowy zbiór dla zadanego identyfikatora transakcji.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart_IdTransakcji	SysCzasKoniec_IdTransakcji
1	Jan	Kowalski	Warszawa	1 034	18 446 744 073 709 551 615

Rys.15. Wersja rekordu dla zadanego identyfikatora transakcji, źródło: opracowanie własne

Predykat temporalny FROM TO zwraca zbiór rekordów temporalnych z zadanego przedziału czasowego. Jednakże w wynikowym zbiorze rekordów nie uwzględnia rekordów, które są aktualne dla chwili czasowej równej górnej granicy czasookresu [2,8,11]. Poniżej przedstawiono przykład zapytania z użyciem klauzuli FROM TO:

```
SELECT * FROM pracownik FOR SYSTEM_TIME FROM '2022-02-  
24 19:35:00' TO '2022-02-25 19:09:39.611159';
```

Na rysunku 16 przedstawiono rezultat działania powyższego zapytania.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart	SysCzasKoniec
1	Jan	Kowalski	Warszawa	2022-02-24 19:37:43.648392	2022-02-25 19:09:39.611159
2	Katarzyna	Nowakowska	Olkusz	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
3	Adam	Grzegorzczak	Gdynia	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
4	Alicja	Jarebska	Jelenia Góra	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
5	Julia	Widawska	Kraków	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999

Rys.16. Wynik działania zapytania na tabeli temporalnej z użyciem predykatu FROM TO, źródło: opracowanie własne

Poniżej przedstawiono przykład zapytania z użyciem klauzuli FROM TO w odniesieniu do tabeli temporalnej z jawnymi znacznikami składającymi identyfikatory transakcji:

```
SELECT * FROM pracownik_trn FOR SYSTEM_TIME FROM  
TRANSACTION 1034 TO TRANSACTION 1038;
```

Na rysunku 17 przedstawiono rezultat działania powyższego zapytania.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart_IdTransakcji	SysCzasKoniec_IdTransakcji
1	Jan	Kowalski	Warszawa	1 034	1 048

Rys.17. Wynik działania zapytania na tabeli temporalnej z znacznikami składającymi identyfikatory transakcji, z użyciem predykatu FROM TO, źródło: opracowanie własne

Predykat temporalny BETWEEN AND zwraca zbiór rekordów temporalnych z zadanego przedziału czasowego. W wynikowym zbiorze rekordów uwzględnia także rekordy, które są aktualne dla chwili czasowej równej dolnej i górnej granicy czasookresu [2,8,11]. Poniżej przedstawiono przykład zapytania z użyciem klauzuli BETWEEN AND:

```
SELECT * FROM pracownik FOR SYSTEM_TIME BETWEEN '2022-02-24 19:35:00' AND '2022-02-25 19:09:39.611159';
```

Na rysunku 18 przedstawiono rezultat działania powyższego zapytania. Zwracany zestaw wyników uwzględnia aktualizację i usunięcie rekordu.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart	SysCzasKoniec
1	Jan	Kowalski	Warszawa	2022-02-24 19:37:43.648392	2022-02-25 19:09:39.611159
1	Jan	Kowalski	Sanok	2022-02-25 19:09:39.611159	2022-02-25 19:32:03.010730
2	Katarzyna	Nowakowska	Olkusz	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
3	Adam	Grzegorzcyk	Gdynia	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
4	Alicja	Jarebska	Jelenia Góra	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
5	Julia	Widawska	Kraków	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999

Rys.18. Wynik działania zapytania na tabeli temporalnej z użyciem predykatu BETWEEN AND, źródło: opracowanie własne

Poniżej przedstawiono przykład zapytania z użyciem klauzuli BETWEEN AND w odniesieniu do tabeli temporalnej z jawnymi znacznikami składającymi identyfikatory transakcji:

```
SELECT * FROM pracownik_trn FOR SYSTEM_TIME BETWEEN TRANSACTION 1034 AND TRANSACTION 1038;
```

Na rysunku 19 przedstawiono rezultat działania tego zapytania.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart_IdTransakcji	SysCzasKoniec_IdTransakcji
1	Jan	Kowalski	Warszawa	1 034	1 048
2	Katarzyna	Nowakowska	Olkusz	1 038	18 446 744 073 709 551 615

Rys.19. Wynik działania zapytania na tabeli temporalnej ze znacznikami składającymi identyfikatory transakcji, z użyciem predykatu BETWEEN AND, źródło: opracowanie własne

Poniżej przedstawiono kilka przykładów zapytań temporalnych zwracających rekordy historyczne, których wartość była aktualna w zadanym przedziale czasowym. Odpowiednio w ciągu ostatniej godziny, ostatniego dnia, ostatniego tygodnia, ostatniego miesiąca i ostatniego roku.

```
SELECT * FROM pracownik FOR SYSTEM_TIME BETWEEN (NOW() - INTERVAL 1 HOUR) AND NOW();
```

```
SELECT * FROM pracownik FOR SYSTEM_TIME BETWEEN (NOW() - INTERVAL 1 DAY) AND NOW();
```

```
SELECT * FROM pracownik FOR SYSTEM_TIME BETWEEN (NOW() - INTERVAL 1 WEEK) AND NOW();
```

```
SELECT * FROM pracownik FOR SYSTEM_TIME BETWEEN (NOW() - INTERVAL 1 MONTH) AND NOW();
```

```
SELECT * FROM pracownik FOR SYSTEM_TIME BETWEEN (NOW() - INTERVAL 1 YEAR) AND NOW();
```

Rozszerzenie temporalne ALL nie wymaga podania żadnego argumentu. Zwraca wszystkie rekordy, bieżące i historyczne [11]. Poniżej przedstawiono przykład zapytania temporalnego z użyciem predykatu ALL na tabeli z jawnymi znacznikami czasowymi:

```
SELECT * FROM pracownik FOR SYSTEM_TIME ALL;
```

Na rysunku 20 przedstawiono rezultat działania powyższego zapytania. Zwraca ono postać bieżącą rekordów oraz wszystkie historyczne wersje rekordów. Wynikowy zbiór rekordów uwzględnia fakt usunięcia rekordu 5 oraz dodanie do tabeli w późniejszym czasie rekordu 6.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart	SysCzasKoniec
1	Jan	Kowalski	Warszawa	2022-02-24 19:37:43.648392	2022-02-25 19:09:39.611159
1	Jan	Kowalski	Sanok	2022-02-25 19:09:39.611159	2022-02-25 19:32:03.010730
2	Katarzyna	Nowakowska	Olkusz	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
3	Adam	Grzegorzcyk	Gdynia	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
4	Alicja	Jarebska	Jelenia Góra	2022-02-24 19:38:07.838646	2038-01-19 04:14:07.999999
5	Julia	Widawska	Kraków	2022-02-24 19:38:07.838646	2022-05-28 19:59:40.465953
6	Sara	Kwiatkowska	Zgierz	2022-07-08 16:01:57.690859	2038-01-19 04:14:07.999999

Rys.20. Wynik działania zapytania na tabeli temporalnej z użyciem predykatu ALL, źródło: opracowanie własne

Poniżej przedstawiono przykład zapytania temporalnego z użyciem predykatu ALL na tabeli temporalnej ze znacznikami, w postaci identyfikatorów transakcji:

```
SELECT * FROM pracownik_trn FOR SYSTEM_TIME ALL;
```

Na rysunku 21 przedstawiono wynik działania powyższego zapytania na tabeli temporalnej po jej modyfikacji.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart_IdTransakcji	SysCzasKoniec_IdTransakcji
1	Jan	Kowalski	Warszawa	1 034	1 048
1	Jan	Kowalski	Sanok	1 048	18 446 744 073 709 551 615
2	Katarzyna	Nowakowska	Olkusz	1 038	18 446 744 073 709 551 615
3	Adam	Grzegorzcyk	Gdynia	1 043	18 446 744 073 709 551 615
4	Alicja	Jarebska	Jelenia Góra	1 043	18 446 744 073 709 551 615
5	Julia	Widawska	Kraków	1 043	18 446 744 073 709 551 615

Rys.21. Wynik działania zapytania na tabeli temporalnej z użyciem predykatu ALL, z widocznymi znacznikami składającymi identyfikatory transakcji, źródło: opracowanie własne

Na rysunku 22 przedstawiono rezultat działania zapytania temporalnego z użyciem predykatu ALL na tabeli temporalnej po usunięciu rekordu. Kolumnie końca przedziału czasowego dla usuniętego rekordu została przypisana nowa wartość identyfikatora transakcji.

IdPracownika	Imie	Nazwisko	Miejscowosc	SysCzasStart_IdTransakcji	SysCzasKoniec_IdTransakcji
1	Jan	Kowalski	Warszawa	1 034	1 048
1	Jan	Kowalski	Sanok	1 048	1 053
2	Katarzyna	Nowakowska	Olkusz	1 038	18 446 744 073 709 551 615
3	Adam	Grzegorzcyk	Gdynia	1 043	18 446 744 073 709 551 615
4	Alicja	Jarebska	Jelenia Góra	1 043	18 446 744 073 709 551 615
5	Julia	Widawska	Kraków	1 043	18 446 744 073 709 551 615

Rys.22. Zawartość tabeli temporalnej po usunięciu rekordu z widocznymi znacznikami składającymi identyfikatory transakcji, źródło: opracowanie własne

Podsumowanie

Artykuł ten dotyczy przetwarzania danych temporalnych dla wymiaru czasu transakcyjnego w środowisku MariaDB. Platforma MariaDB obsługuje oba wymiary czasu, zarówno czas transakcyjny i czas rzeczywisty. Dzięki temu możliwa jest także obsługa tabel bitemporalnych.

Dodanie rekordu do tabeli temporalnej przebiega w identyczny sposób jak dla klasycznej tabeli. Każdy nowo dodany rekord jest rekordem bieżącym (stanowi jego aktualną wersję). W przypadku aktualizacji i usuwania danych w tabeli temporalnej wersjonowanej systemowo z obsługą wymiaru czasu transakcyjnego, modyfikacji podlegać mogą tylko i wyłącznie rekordy bieżące. Niedopuszczalne jest modyfikowanie i usuwanie rekordów temporalnych. Użytkownik nie może także ingerować w żaden sposób, w wartości składowane, w kolumnach znaczników czasowych, które są generowane automatycznie przez system. Każda modyfikacja danych bieżących skutkuje utworzeniem rekordu temporalnego dla każdego aktualizowanego lub usuwanego rekordu.

Składowanie wszystkich wersji danych (całej historii zmian) skutkuje dużym przyrostem danych. Tabela temporalna z upływem czasu, coraz bardziej zwiększać będzie swój rozmiar. Środowisko MariaDB udostępnia kilka sposobów usunięcia danych temporalnych z tabeli temporalnej wersjonowanej systemowo [14]. Należy zaznaczyć, że usunięcie danych historycznych wiąże się z częściową lub całkowitą utratą tych danych, a więc z utratą historii. Jest to niezgodne z założeniami standardu

SQL:2011. W celu uniknięcia przypadkowego usunięcia danych z tabeli temporalnej wymagane jest posiadanie uprawnień DELETE HISTORY. Rozwiązanie to pozwala chronić integralność danych temporalnych.

Usunięcie rekordów temporalnych możliwe jest poprzez usunięcie i ponowne przywrócenie wersjonowania tabeli temporalnej. Usuwana jest wówczas cała historia zmian. Jest to proces czasochłonny, gdyż wiąże się to z przebudowaniem całej tabeli temporalnej na nowo. Możliwe także jest fragmentaryczne usunięcie danych temporalnych poprzez usunięcie wybranych partycji historycznych. Przy czym, niemożliwe będzie usunięcie partycji historycznej, jeżeli jest to jedyna istniejąca partycja historyczna w tabeli temporalnej. Jeszcze innym wariantem usunięcia danych temporalnych jest użycie w poleceniu DELETE klauzuli HISTORY. Rozwiązanie to pozwala usunąć wszystkie rekordy temporalne lub też usunąć rekordy temporalne do określonego momentu w czasie, czy też do konkretnej transakcji (o podanym numerze). Usunięcie danych temporalnych za pomocą polecenia DELETE HISTORY z klauzulą BEFORE SYSTEM_TIME z wartością większą niż ROW_END dla rekordów bieżących skutkuje usunięciem rekordów temporalnych, a rekordy aktualne przenoszone zostają do obszaru składowania rekordów temporalnych. We wcześniejszych wersjach serwera MariaDB możliwe było także użycie polecenia TRUNCATE TABLE do usunięcia danych temporalnych. Od wersji 10.4.5 niemożliwe jest użycie tej instrukcji na tabelach wersjonowanych systemowo, w celu zachowania zgodności ze standardem SQL:2011.

Zapytanie o dane temporalne możliwe jest do zrealizowania przy użyciu odpowiednich predykatów temporalnych pozwalających pobierać dane na określony moment w czasie lub też dane, które były aktualne w określonym przedziale czasu. Na platformie MariaDB zaimplementowano także jeden dodatkowy, ponadstandardowy predykat temporalny ALL, który zwraca zarówno rekordy aktualne jak i wszystkie poprzednie wersje rekordów [2,8,11].

W kwestii zgodności z wytycznymi zawartymi w standardzie SQL:2011, odnośnie składowania i obsługi danych temporalnych dla wymiaru czasu transakcyjnego środowisko MariaDB zapewnia możliwość tworzenia i obsługi tabel wersjonowanych systemowo, zarówno z jawnymi jak i niejawnymi znacznikami czasowymi. Udostępnia także predykaty czasowe umożliwiające temporalne przetwarzanie danych składowanych, w tabelach temporalnych wersjonowanych systemowo. Ponadto rozszerza ten zbiór o dodatkowy predykat ALL w stosunku do zapisu standardu SQL:2011. Platforma MariaDB obsługuje większość typów danych istotnych dla przetwarzania danych temporalnych, które są uwzględnione w standardzie SQL:2011 oprócz typu danych INTERVAL.

W odniesieniu do innych relacyjnych baz danych pod kątem implementacji predykatów temporalnych do przetwarzania danych temporalnych, platforma MariaDB wypada korzystniej od serwera Oracle, gdzie nie zaimplementowano predykatów FROM TO oraz ALL [4,15,17]. Natomiast środowisko MS SQL Server udostępnia jeszcze jeden ponadstandardowy predykat CONTAINED IN niedostępny na platformie MariaDB [3,16,17]. Pozwala on zwracać zbiór rekordów otwartych lub zamkniętych w przedziale czasu ograniczonym wartościami parametryzującymi ten predykat czasowy.

Artykuł ten poświęcony został przetwarzaniu danych temporalnych na platformie MariaDB ze szczególnym uwzględnieniem rejestracji zmian w wymiarze czasu transakcyjnego. Kolejnym etapem pracy będzie zbadanie stopnia realizacji założeń temporalnych rozszerzeń standardu SQL:2011, na platformie MariaDB w kontekście obsługi danych temporalnych, dla wymiaru czasu rzeczywistego oraz prezentacja zaimplementowanych funkcjonalności temporalnych.

Autorzy: dr inż. Sebastian Łacheciński, Uniwersytet Łódzki, Instytut Logistyki i Informatyki, Katedra Informatyki Ekonomicznej, ul. Rewolucji 1905 r. 37, 90-214 Łódź, E-mail: sebastian.lachecinski@uni.lodz.pl

LITERATURA

- [1] Łacheciński S., Modelowanie danych temporalnych w relacyjnym modelu danych, *Informatyka Ekonomiczna*, 4(46) (2017), 90-107
- [2] Łacheciński S., Składowanie i przetwarzanie danych temporalnych w świetle wymagań standardu języka SQL ISO-IEC 9075, *Przegląd Elektrotechniczny*, 96 (2020), nr 10, 184-191
- [3] Łacheciński S., Obsługa danych temporalnych na platformie MS SQL Server i Azure SQL Database, *Przegląd Elektrotechniczny*, 96 (2020), nr 12, 95-101
- [4] Łacheciński S., Obsługa danych temporalnych dla wymiaru czasu transakcyjnego na platformie ORACLE, *Przegląd Elektrotechniczny*, 97 (2021), nr 11, 159-163
- [5] Łacheciński S., Obsługa danych temporalnych dla wymiaru czasu rzeczywistego na platformie ORACLE, *Przegląd Elektrotechniczny*, 97 (2021), nr 12, 86-91
- [6] Łacheciński S., Składowanie danych temporalnych dla wymiaru czasu transakcyjnego na platformie MariaDB, *Przegląd Elektrotechniczny*, 98 (2022), nr 10, 28-33
- [7] Date C.J., Darwen H., Lorentzos N., *Time and relational theory Temporal Databases in the Relational Model and SQL*, 2014, Morgan Kaufmann
- [8] Kulkarni K., Jan-Eike Michels, 2012, *Temporal features in SQL:2011*, 34-43
<https://sigmodrecord.org/publications/sigmodRecord/1209/pdfs/07.industry.kulkarni.pdf>
- [9] Temporal extension SQL:
https://link.springer.com/content/pdf/10.1007%2F978-1-4899-7993-3_80729-1.pdf
- [10] MariaDB Temporal Tables:
<https://mariadb.com/kb/en/temporal-tables/>
- [11] MariaDB Querying Historical Data:
<https://mariadb.com/kb/en/system-versioned-tables/#querying-historical-data>
- [12] MariaDB Creating a System-Versioned Table:
<https://mariadb.com/kb/en/system-versioned-tables/#creating-a-system-versioned-table>
- [13] MariaDB Transaction-Precise History in InnoDB:
<https://mariadb.com/kb/en/system-versioned-tables/#transaction-precise-history-in-innodb>
- [14] MariaDB Removing Old History:
<https://mariadb.com/kb/en/system-versioned-tables/#removing-old-history>
- [15] Oracle Using Oracle Flashback Version Query:
https://docs.oracle.com/database/121/ADFNS/adfn_flashback.htm#ADFNS01004
- [16] MS SQL Server Querying data in a system-versioned temporal table: <https://docs.microsoft.com/en-us/sql/relational-databases/tables/querying-data-in-a-system-versioned-temporal-table?view=sql-server-ver16>
- [17] Petković D., Support of Temporal Data in Database Systems, *International Journal of Computer Applications* (0975 –8887), Volume 152 –No.4, October 2016, 26-33