

## Składowanie danych temporalnych dla wymiaru czasu rzeczywistego na platformie MariaDB

**Streszczenie.** Celem artykułu jest wskazanie zakresu implementacji obsługi danych temporalnych dla czasu rzeczywistego 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 składowania 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 storage of temporal data by MariaDB environment. (**Storage of temporal data for the valid time on the MariaDB platform**).

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

**Keywords:** temporal databases, temporal tables, temporal operators, valid 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 i przetwarzaniu danych temporalnych w wymiarze czasu transakcyjnego na platformie MariaDB [6,7]. Artykuł ten jest kontynuacją rozważań nt. możliwości składowania i przetwarzania danych temporalnych oferowanych przez serwery SQL. W całości poświęcony został on składowaniu danych temporalnych na platformie MariaDB z uwzględnieniem wymiaru czasu rzeczywistego. Przeprowadzona została analiza zakresu możliwości magazynowania danych temporalnych dostępnych na platformie MariaDB, dla czasu rzeczywistego. 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 tworzenia tabel wersjonowanych aplikacyjnie składujących dane temporalne dla wymiaru czasu rzeczywistego 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 [8-10]:

- 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 [11].

Dane temporalne z uwzględnieniem wymiaru czasu rzeczywistego składowane są w tabelach temporalnych wersjonowanych aplikacyjnie. Tabele wersjonowane aplikacyjnie dla obsługi czasu rzeczywistego przechowują dodatkowo wartość czasookresu, w którym dany rekord był prawdziwy (był aktualny w modelowanym świecie rzeczywistym). Wartości czasookresu przechowywane są w dwóch odrębnych kolumnach składujących odpowiednio początek i koniec zdefiniowanego przez użytkownika czasookresu. Kolumny te muszą posiadać ten sam typ danych. W środowisku MariaDB dopuszczalne jest użycie następujących systemowych typów danych, w odniesieniu do kolumn składujących granice czasookresu: DATE, TIMESTAMP, DATETIME. Typ DATETIME jest dodatkowym typem danych dopuszczonym przez środowisko MariaDB do definiowania kolumn czasookresu, który nie został zdefiniowany w zapisie standardu SQL:2011. Wykaz dostępnych na platformie MariaDB systemowych typów danych z kategorii data i czas przedstawiono w pracy [6]. Niedopuszczalne jest użycie wartości NULL, w odniesieniu do kolumn składujących granice czasookresu. W sposób niejawni do tabeli temporalnej wersjonowanej aplikacyjnie, dodawane jest także ograniczenie sprawdzające, czy wartość początku czasookresu jest mniejsza od wartości końca czasookresu. Nazwa tego warunku ograniczającego rozpoczyna się od nazwy czasookresu określonej przez użytkownika na etapie tworzenia definicji tabeli temporalnej wersjonowanej aplikacyjnie. Od wersji 10.5 serwera MariaDB dostępna jest klauzula WITHOUT OVERLAPS, która nie dopuszcza, aby dla kolejnych rekordów, tego samego klucza czasookresy mogły się pokrywać.

Tabele wersjonowane aplikacyjnie przydatne są w zastosowaniach, gdzie istotna jest wiedza na temat tego, kiedy dany rekord był rekordem aktualnym (w jakim okresie czasu), do kiedy był rekordem aktualnym, czy też od kiedy stanie się rekordem aktualnym. Jako przykład można podać rejestrację zmian cen towarów w czasie, śledzenie historii zatrudnienia pracownika w firmie, jego kolejne awanse, jak

również migracje pomiędzy poszczególnymi działami danej organizacji, czy też datę rozpoczęcia wznowienia kolejnego ubezpieczenia. W odróżnieniu od tabel wersjonowanych systemowo to użytkownik jest odpowiedzialny za zdefiniowanie wartości granic dla czasookresu, w którym dany rekord był rekordem aktualnym.

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

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

Wymagania	SQL:2011	MariaDB 10.6.5
Definicja czasookresu	✓	✓
Typ danych PERIOD	✓	-
Tabele wersjonowane aplikacyjnie	✓	✓
Implementacja predykatów czasowych TWA <sup>1</sup>	✓	-
Tabele bitemporalne	✓	✓
Aktualizacja i usuwanie rekordów z zadanego przedziału czasowego	✓	✓
Jawne znaczniki czasowe	✓	✓
Niejawne znaczniki czasowe	✓	-
Temporalne ograniczenie PK	✓	✓
Temporalne ograniczenie FK REFERENCES	✓	-
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 rzeczywistego. Możliwe jest tworzenie temporalnych tabel wersjonowanych aplikacyjnie, ale tylko z jawnymi znacznikami czasu. Obsługiwane jest aktualizowanie i usuwanie rekordów z zadanego przedziału czasowego. Zaimplementowane zostało w postaci indeksu ograniczającego także temporalne ograniczenie wykluczające nakładanie się czasookresów. Indeks taki musi być kluczem głównym tabeli temporalnej lub indeksem unikatowym. Nie zaimplementowane zostało temporalne ograniczenie integralności referencyjnej oraz typ danych PERIOD. Środowisko MariaDB udostępnia także trzy z czterech typów danych zdefiniowanych w standardzie SQL:2011, oprócz typu INTERVAL. Nie zaimplementowane zostały także predykaty czasowe do obsługi zapytań na tabelach wersjonowanych aplikacyjnie. Odpytywanie tych tabel możliwe jest w tradycyjny sposób jak dla klasycznych tabel. Tabele temporalne do obsługi czasu rzeczywistego w środowisku MariaDB mogą obsługiwać tylko jeden wymiar czasu rzeczywistego.

### Tworzenie tabeli temporalnej z obsługą czasu rzeczywistego w MariaDB

Sposoby tworzenia tabeli temporalnej wersjonowanej aplikacyjnie w środowisku MariaDB [12-14]:

- tabela temporalna z opcją PERIOD FOR
- adaptacja istniejącej tabeli do tabeli temporalnej wersjonowanej aplikacyjnie przez dodanie do niej definicji czasookresu

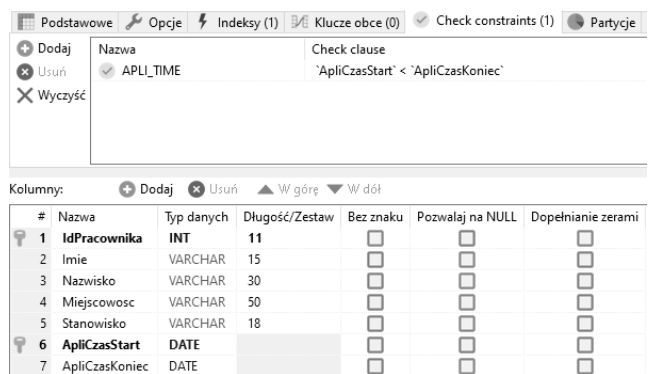
<sup>1</sup> TWA-tabela wersjonowana aplikacyjnie

- adaptacja istniejącej tabeli do tabeli temporalnej wersjonowanej aplikacyjnie przez dodanie do niej definicji czasookresu wraz z kolumnami składającymi wartości graniczne czasookresu
- tabela temporalna z wykluczeniem nakładania się okresów czasowych, z użyciem klauzuli WITHOUT OVERLAPS

Poniżej przedstawiono przykład tworzenia tabeli temporalnej wersjonowanej aplikacyjnie z jawnie definiowanymi znacznikami czasowymi:

```
CREATE TABLE pracownik_vt
(
  IdPracownika int NOT NULL,
  Imie VARCHAR(15) NOT NULL,
  Nazwisko VARCHAR(30) NOT NULL,
  Miejscowosc VARCHAR(50) NOT NULL,
  Stanowisko VARCHAR(18) NOT NULL,
  PRIMARY KEY(IdPracownika,ApliCzasStart),
  ApliCzasStart DATE NULL,
  ApliCzasKoniec DATE NULL,
  PERIOD FOR APLI_TIME (ApliCzasStart,ApliCzasKoniec)
);
```

Na rysunku 1 przedstawiono strukturę utworzonej tabeli temporalnej wersjonowanej aplikacyjnie, z jawnie definiowanymi znacznikami czasowymi. Ponadto uwidoczniona została także definicja warunku ograniczającego dodanego niejawnie do tworzonej tabeli temporalnej. Zadaniem tego warunku jest wymuszenie, aby wartość początku przedziału czasowego była mniejsza od wartości końca przedziału czasowego dla zdefiniowanego czasookresu.



Rys.1. Struktura tabeli temporalnej wersjonowanej aplikacyjnie z jawnie definiowanymi znacznikami czasowymi, Źródło: opracowanie własne

Kolejny przykład stanowi ilustrację tworzenia tabeli temporalnej wersjonowanej aplikacyjnie poprzez modyfikację istniejącej już tabeli do tabeli temporalnej. Jeżeli w istniejącej tabeli, pola definiujące znaczniki czasu dla dodawanego czasookresu dopuszczają wartości NULL, zostaną one niejawnie zmienione na NOT NULL:

```
ALTER TABLE pracownik_vt
ADD PERIOD FOR APLI_TIME
(ApliCzasStart,ApliCzasKoniec);
```

Poniżej przedstawiono przykład tworzenia tabeli temporalnej wersjonowanej aplikacyjnie poprzez modyfikację istniejącej tabeli do tabeli temporalnej, z jednoczesną weryfikacją, czy podany czasookres już istnieje. Jeżeli nie istnieje, zostanie on utworzony podczas zmiany definicji istniejącej tabeli do tabeli temporalnej. Znaczniki czasookresu określane są w sposób jawny.

```
ALTER TABLE pracownik_vt
ADD PERIOD IF NOT EXISTS FOR APLI_TIME
(ApliCzasStart,ApliCzasKoniec);
```

Następny przykład ilustruje tworzenie tabeli temporalnej wersjonowanej aplikacyjnie poprzez modyfikację definicji istniejącej tabeli do tabeli temporalnej, poprzez dodanie kolumn znaczników czasookresu oraz zdefiniowanie czasookresu:

```
ALTER TABLE pracownik_vt
ADD COLUMN ApliCzasStart DATE NOT NULL,
ADD COLUMN ApliCzasKoniec DATE NOT NULL,
ADD PERIOD FOR APLI_TIME
(ApliCzasStart,ApliCzasKoniec);
```

### Temporalne ograniczenie klucza podstawowego

W środowisku MariaDB możliwe jest także utworzenie tabeli temporalnej wersjonowanej aplikacyjnie z dodatkowym ograniczeniem wykluczającym nakładanie się czasookresów. Dzięki temu niedopuszczalne będzie zajmowanie przez pracownika dwóch różnych stanowisk w tym samym czasie.

Poniżej przedstawiono przykład tworzenia tabeli temporalnej wersjonowanej aplikacyjnie z uwzględnieniem wykluczenia nakładania się czasookresów poprzez użycie indeksu ograniczającego. Indeks taki musi być indeksem unikatowym bądź też kluczem podstawowym tworzonej tabeli temporalnej:

```
CREATE TABLE pracownik_vt_wo
(
IdPracownika int PRIMARY KEY NOT NULL,
Imie VARCHAR(15) NOT NULL,
Nazwisko VARCHAR(30) NOT NULL,
Miejscowosc VARCHAR(50) NOT NULL,
Stanowisko VARCHAR(18) NOT NULL,
ApliCzasStart DATE NOT NULL,
ApliCzasKoniec DATE NOT NULL,
PERIOD FOR APLI_TIME (ApliCzasStart,ApliCzasKoniec),
UNIQUE (IdPracownika, Stanowisko, APLI_TIME WITHOUT
OVERLAPS)
);
```

Na rysunku 2 przedstawiono strukturę utworzonej tabeli temporalnej wersjonowanej aplikacyjnie z jawnie definiowanymi znacznikami czasowymi z wykluczeniem nakładania się czasookresów. Widoczna jest także struktura indeksu klucza podstawowego tabeli temporalnej oraz ograniczającego indeksu unikatowego założonego na kombinacji pól: klucza podstawowego, stanowiska i czasookresu. Kombinacja taka gwarantuje, że ten sam pracownik w określonym przedziale czasu nie będzie mógł piastować różnych stanowisk.

#	Nazwa	Typ danych	Długość/Zestaw	Bez znaku	Pozwalaj na NULL	Dope
1	IdPracownika	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	
2	Imie	VARCHAR	15	<input type="checkbox"/>	<input type="checkbox"/>	
3	Nazwisko	VARCHAR	30	<input type="checkbox"/>	<input type="checkbox"/>	
4	Miejscowosc	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	
5	Stanowisko	VARCHAR	18	<input type="checkbox"/>	<input type="checkbox"/>	
6	ApliCzasStart	DATE		<input type="checkbox"/>	<input type="checkbox"/>	
7	ApliCzasKoniec	DATE		<input type="checkbox"/>	<input type="checkbox"/>	

Rys.2. Struktura tabeli temporalnej wersjonowanej aplikacyjnie, z wykluczeniem nakładania się czasookresów, implementowanym za pomocą ograniczającego indeksu unikatowego uwzględniającego w swojej strukturze pole klucza podstawowego, źródło: opracowanie własne

Alternatywę dla tego rozwiązania stanowi utworzenie tabeli temporalnej wersjonowanej aplikacyjnie bez zdefiniowania ograniczenia klucza podstawowego. Poniżej przedstawiono przykład tworzenia tabeli temporalnej z wykluczaniem nakładania czasookresu implementowanej tylko za pomocą indeksu ograniczającego, bez zdefiniowania dla niej klucza podstawowego:

```
CREATE TABLE pracownik_vt_wo
(
IdPracownika int NOT NULL,
Imie VARCHAR(15) NOT NULL,
Nazwisko VARCHAR(30) NOT NULL,
Miejscowosc VARCHAR(50) NOT NULL,
Stanowisko VARCHAR(18) NOT NULL,
ApliCzasStart DATE NOT NULL,
ApliCzasKoniec DATE NOT NULL,
PERIOD FOR APLI_TIME (ApliCzasStart,ApliCzasKoniec),
UNIQUE (IdPracownika, Stanowisko, APLI_TIME WITHOUT
OVERLAPS)
);
```

Na rysunku 3 przedstawiono strukturę utworzonej tabeli temporalnej wersjonowanej aplikacyjnie z jawnie definiowanymi znacznikami czasowymi z wykluczeniem nakładania się czasookresów implementowanym przy użyciu samego indeksu ograniczającego.

#	Nazwa	Typ danych	Długość/Zestaw	Bez znaku	Pozwalaj na NULL	Dope
1	IdPracownika	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	
2	Imie	VARCHAR	15	<input type="checkbox"/>	<input type="checkbox"/>	
3	Nazwisko	VARCHAR	30	<input type="checkbox"/>	<input type="checkbox"/>	
4	Miejscowosc	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	
5	Stanowisko	VARCHAR	18	<input type="checkbox"/>	<input type="checkbox"/>	
6	ApliCzasStart	DATE		<input type="checkbox"/>	<input type="checkbox"/>	
7	ApliCzasKoniec	DATE		<input type="checkbox"/>	<input type="checkbox"/>	

Rys.3. Struktura tabeli temporalnej wersjonowanej aplikacyjnie, z wykluczeniem nakładania się czasookresów, implementowanym tylko za pomocą ograniczającego indeksu unikatowego, bez użycia klucza podstawowego, źródło: opracowanie własne

Możliwa jest także bezpośrednia implementacja ograniczenia temporalnego klucza głównego, podczas jego definiowania na etapie tworzenia tabeli temporalnej. Przykład realizacji tego podejścia przedstawiono poniżej:

```
CREATE TABLE pracownik_vt_wo
(
IdPracownika int NOT NULL,
Imie VARCHAR(15) NOT NULL,
Nazwisko VARCHAR(30) NOT NULL,
Miejscowosc VARCHAR(50) NOT NULL,
Stanowisko VARCHAR(18) NOT NULL,
ApliCzasStart DATE NOT NULL,
ApliCzasKoniec DATE NOT NULL,
PERIOD FOR APLI_TIME (ApliCzasStart,ApliCzasKoniec),
PRIMARY KEY (IdPracownika, Stanowisko,APLI_TIME
WITHOUT OVERLAPS)
);
```

Na rysunku 4 przedstawiono strukturę utworzonej tabeli temporalnej wersjonowanej aplikacyjnie z jawnie definiowanymi znacznikami czasowymi, z wykluczeniem nakładania się czasookresów. Widoczna jest także struktura indeksu klucza podstawowego tabeli temporalnej składająca się z pól: IdPracownika, Stanowisko oraz pól zdefiniowanego czasookresu. W odróżnieniu od poprzedniej konstrukcji nie jest tworzony dodatkowo,

ograniczający indeks unikatowy na tych polach. Samo ograniczenie klucza podstawowego spełnia wymóg unikatowości.

#	Nazwa	Typ danych	Długość/Zestaw	Bez znaku	Pozwalaj na NULL
1	IdPracownika	INT	11	<input type="checkbox"/>	<input type="checkbox"/>
2	Imie	VARCHAR	15	<input type="checkbox"/>	<input type="checkbox"/>
3	Nazwisko	VARCHAR	30	<input type="checkbox"/>	<input type="checkbox"/>
4	Miejscowosc	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>
5	Stanowisko	VARCHAR	18	<input type="checkbox"/>	<input type="checkbox"/>
6	ApliCzasStart	DATE		<input type="checkbox"/>	<input type="checkbox"/>
7	ApliCzasKoniec	DATE		<input type="checkbox"/>	<input type="checkbox"/>

Rys.4. Struktura tabeli temporalnej wersjonowanej aplikacyjnie, z wykluczeniem nakładania się czasookresów, z uwzględnieniem klauzuli WITHOUT OVERLAPS w temporalnym ograniczeniu klucza podstawowego, źródło: opracowanie własne

### Tworzenie tabeli bitemporalnej w MariaDB

W środowisku MariaDB od wersji 10.4.3, po wprowadzeniu obsługi czasu rzeczywistego możliwe także stało się tworzenie tabel bitemporalnych obsługujących oba wymiary czasu: transakcyjnego i rzeczywistego [15].

Poniżej przedstawiono przykład utworzenia tabeli bitemporalnej, z uwzględnieniem wykluczania nakładania się okresów czasowych, implementowanym bezpośrednio, jako temporalne ograniczenie klucza podstawowego. Rejestruje ona nie tylko czas rozpoczęcia i zakończenia zatrudnienia pracownika na danym stanowisku, ale także czas utrwalenia tych danych w tabeli temporalnej. Wprowadzenie danych do systemu nie musi koniecznie pokrywać się z datą zatrudnienia pracownika.

```
CREATE TABLE pracownik_bt
(
  IdPracownika int NOT NULL,
  Imie VARCHAR(15) NOT NULL,
  Nazwisko VARCHAR(30) NOT NULL,
  Miejscowosc VARCHAR(50) NOT NULL,
  Stanowisko VARCHAR(18) NOT NULL,
  ApliCzasStart DATE NULL,
  ApliCzasKoniec DATE NULL,
  SysCzasStart TIMESTAMP(6) AS ROW START INVISIBLE,
  SysCzasKoniec TIMESTAMP(6) AS ROW END INVISIBLE,
  PERIOD FOR APLI_TIME (ApliCzasStart,ApliCzasKoniec),
  PERIOD FOR SYSTEM_TIME(SysCzasStart,SysCzasKoniec),
  PRIMARY KEY (IdPracownika,APLI_TIME WITHOUT OVERLAPS)
)
WITH SYSTEM VERSIONING;
```

#	Nazwa	Typ danych	Długość/Zestaw	Bez znaku	Pozwalaj na NULL	Wyrażenie	Wirtualność
1	IdPracownika	INT	11	<input type="checkbox"/>	<input type="checkbox"/>		
2	Imie	VARCHAR	15	<input type="checkbox"/>	<input type="checkbox"/>		
3	Nazwisko	VARCHAR	30	<input type="checkbox"/>	<input type="checkbox"/>		
4	Miejscowosc	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>		
5	Stanowisko	VARCHAR	18	<input type="checkbox"/>	<input type="checkbox"/>		
6	ApliCzasStart	DATE		<input type="checkbox"/>	<input type="checkbox"/>		
7	ApliCzasKoniec	DATE		<input type="checkbox"/>	<input type="checkbox"/>		
8	SysCzasStart	TIMESTA...	6	<input type="checkbox"/>	<input type="checkbox"/>	ROW START	stored
9	SysCzasKoniec	TIMESTA...	6	<input type="checkbox"/>	<input type="checkbox"/>	ROW END	stored

Rys.5. Struktura tabeli bitemporalnej wersjonowanej aplikacyjnie i systemowo, z wykluczeniem nakładania się okresów czasowych, implementowanym bezpośrednio w temporalnym ograniczeniu klucza podstawowego, źródło: opracowanie własne

Na rysunku 5 przedstawiono strukturę utworzonej tabeli bitemporalnej wersjonowanej aplikacyjnie i systemowo.

### Podsumowanie

Artykuł ten w całości poświęcony został składowaniu danych temporalnych na platformie MariaDB z uwzględnieniem wymiaru czasu rzeczywistego. Dane temporalne składowane są w tabelach wersjonowanych aplikacyjnie. Wartości graniczne czasookresu składowane są w dwóch odrębnych kolumnach, które muszą posiadać ten sam typ danych. Nie są one automatycznie generowane przez system, w odróżnieniu od znaczników czasowych, dla tabel z obsługą wymiaru czasu transakcyjnego. Odpowiedzialność za określenie tych wartości spoczywa na użytkowniku. Kolumny te nie mogą przechowywać wartości NULL. Środowisko MariaDB umożliwi tworzenie tabel wersjonowanych aplikacyjnie tylko z jawnymi znacznikami czasowymi, w odróżnieniu od platformy Oracle, gdzie możliwe również jest utworzenie tabeli temporalnej wersjonowanej aplikacyjnie z obsługą kilku wymiarów czasu rzeczywistego. Funkcjonalność taka jest dostępna na serwerze Oracle [5].

Z punktu widzenia zgodności z wymaganiami zawartymi w zapisie standardu SQL:2011, odnośnie składowania danych temporalnych dla wymiaru czasu rzeczywistego środowisko MariaDB zapewnia możliwość tworzenia tabeli wersjonowanych aplikacyjnie z jawnymi znacznikami czasu, bez możliwości utworzenia takiej tabeli z niejawnymi znacznikami czasu. Realizowana jest także funkcjonalność wykluczania nakładania się okresów czasowych, implementowana poprzez wprowadzenie klauzuli WITHOUT OVERLAPS, w ograniczającym indeksie unikatowym lub w temporalnym ograniczeniu klucza podstawowego tabeli temporalnej wersjonowanej aplikacyjnie. Platforma MariaDB obsługuje wszystkie typy danych specyfikowane w standardzie SQL:2011, poza wyjątkiem typu INTERVAL. Dostępna jest także funkcjonalność temporalnej aktualizacji i usuwania rekordów z zadanego przedziału czasowego. Funkcjonalności takiej brakuje w innych wiodących systemach RDBMS tj. MS SQL Server [3,16-18] czy Oracle [5,16,17,19]. Niedostępne jest ograniczenie temporalnej integralności referencyjnej.

Niezaimplementowane zostały predykaty czasowe dedykowane odpytywaniu tabel temporalnych z obsługą czasu rzeczywistego. Odpytywanie tabel temporalnych odbywa się w standardowy sposób, jak dla klasycznych tabel. Zagadnienia związane z przetwarzaniem danych temporalnych składowanych w tabelach wersjonowanych aplikacyjnie zostaną szerzej zaprezentowane w kolejnym artykule.

Kolejnym etapem pracy będzie zbadanie stopnia realizacji założeń temporalnych standardu SQL:2011, na platformie MariaDB w kontekście przetwarzania danych temporalnych dla wymiaru czasu rzeczywistego, a także 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](mailto: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] Łacheciński S., Przetwarzanie danych temporalnych dla wymiaru czasu transakcyjnego na platformie MariaDB, *Przegląd Elektrotechniczny*, 98 (2022), nr 11, 307-312
- [8] Date C.J., Darwen H., Lorentzos N., *Time and relational theory Temporal Databases in the Relational Model and SQL*, 2014, Morgan Kaufmann
- [9] Kulkarni K., Jan-Eike Michels, 2012, *Temporal features in SQL:2011*, 34-43  
<https://sigmodrecord.org/publications/sigmodRecord/1209/pdfs/07.industry.kulkarni.pdf>
- [10] Temporal extension SQL:  
[https://link.springer.com/content/pdf/10.1007%2F978-1-4899-7993-3\\_80729-1.pdf](https://link.springer.com/content/pdf/10.1007%2F978-1-4899-7993-3_80729-1.pdf)
- [11] MariaDB Temporal Tables:  
<https://mariadb.com/kb/en/temporal-tables/>
- [12] MariaDB Creating tables with time periods:  
<https://mariadb.com/kb/en/application-time-periods/#creating-tables-with-time-periods>
- [13] MariaDB Adding and removing time periods:  
<https://mariadb.com/kb/en/application-time-periods/#adding-and-removing-time-periods>
- [14] MariaDB Without overlaps:  
<https://mariadb.com/kb/en/application-time-periods/#without-overlaps>
- [15] MariaDB Bitemporal tables:  
<https://mariadb.com/kb/en/bitemporal-tables/>
- [16] Petković D., Support of Temporal Data in Database Systems, *International Journal of Computer Applications* (0975 –8887), Volume 152 –No.4, October 2016, 26-33
- [17] Petkovic D., Temporal Data in relational Database Systems: A Comparison, Conference: WorldCIST (1) 2016, At Recife, Volume 1
- [18] MS SQL Server Temporal tables: <https://docs.microsoft.com/pl-pl/sql/relational-databases/tables/temporal-tables?view=sql-server-ver16>
- [19] Temporal Validity in Oracle Database 12c Release 1 (12.1):  
<https://oracle-base.com/articles/12c/temporal-validity-12cr1>