

Składowanie danych temporalnych dla wymiaru czasu transakcyjnego na platformie IBM Db2

Streszczenie. Celem artykułu jest wskazanie zakresu implementacji obsługi danych temporalnych dla czasu transakcyjnego w środowisku IBM Db2 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 IBM Db2.

Abstract. The aim of the article is an indication the scope of the implementation of temporal data support for transaction time in the IBM Db2 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 processing of temporal data by IBM Db2 environment. (Storage of temporal data for the transaction time on the IBM Db2 platform).

Słowa kluczowe: temporalne bazy danych, temporalne tabele, temporalne operatory, czas transakcyjny, SQL:2011, IBM Db2.

Keywords: temporal databases, temporal tables, temporal operators, transaction time, SQL:2011, IBM Db2.

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 temporalne 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 na platformie MariaDB w wymiarze czasu transakcyjnego [6,7] i czasu rzeczywistego [8,9]. Artykuł ten jest kontynuacją rozważań nt. możliwości składowania i przetwarzania danych temporalnych oferowanych przez serwery SQL. W całości poświęcony został on możliwościom składowania danych temporalnych na platformie IBM Db2 z uwzględnieniem wymiaru czasu transakcyjnego. Przeprowadzona została analiza zakresu możliwości magazynowania danych temporalnych dostępnych na platformie IBM Db2, 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 tworzenia tabel wersjonowanych systemowo, składających dane temporalne dla wymiaru czasu transakcyjnego w środowisku IBM Db2.

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 [10-12]:

- 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 IBM Db2

Obsługa danych temporalnych w środowisku IBM Db2 (wówczas IBM DB2) została wprowadzona w wersji 10.1.0, w 2012 roku [13]. Dostępna jest obsługa tabel temporalnych wersjonowanych systemowo dla wymiaru czasu transakcyjnego, a także obsługa tabel temporalnych wersjonowanych aplikacyjnie dla wymiaru czasu rzeczywistego. Na platformie IBM Db2 te wymiary czasu odpowiednio nazywane są czasem systemowym `SYSTEM_TIME` oraz czasem biznesowym `BUSINESS_TIME`. Ponadto możliwe jest także tworzenie i obsługiwanie tabel bitemporalnych obsługujących jednocześnie oba wymiary czasu [14].

Wersje bieżące rekordów składowane są w tabeli temporalnej wersjonowanej systemowo, zaś wersje historyczne rekordów składowane są w powiązanej z nią tabelą historii. Taka konstrukcja pozwala realizować zapytania w oparciu o dane aktualne jak również dane temporalne składowane w tabeli historii powiązanej z tabelą temporalną wersjonowaną systemowo. Definicja tabeli temporalnej z czasem systemowym jest poszerzona w stosunku do klasycznej postaci tabeli o dodatkowe trzy kolumny. Dwie kolumny przechowują odpowiednio wartości początku i końca przedziału czasowego, w którym dany rekord był aktualny. Kolumny te muszą mieć przypisany typ danych `TIMESTAMP`. Niedopuszczalne jest użycie wartości `NULL` w odniesieniu do tych kolumn. Mogą być one definiowane, jako jawne lub niejawne znaczniki czasowe. Domyślnie stosowanym rozwiązaniem są jawne znaczniki czasowe. Niejawne znaczniki czasowe określane są za pomocą opcji `IMPLICITLY HIDDEN`. Kolumna przechowująca początek przedziału czasowego zawiera wartość sygnatury czasu odpowiadającej momentowi utworzenia rekordu. Kolumna przechowująca koniec czasookresu zawiera wartość znacznika czasu odpowiadającego chwili czasowej, w której dany rekord jest modyfikowany lub usuwany (przestał być rekordem bieżącym). Przy czym wartość początkowa jest włączana do czasookresu, natomiast wartość końcowa nie jest uwzględniana w przedziale definiującym granice czasookresu. W oparciu o te dwie kolumny tworzona jest definicja czasookresu. Wartości w kolumnach określających granice czasookresu są automatycznie generowane i utrzymywane przez system. Użytkownik nie może ingerować w te wartości [15].

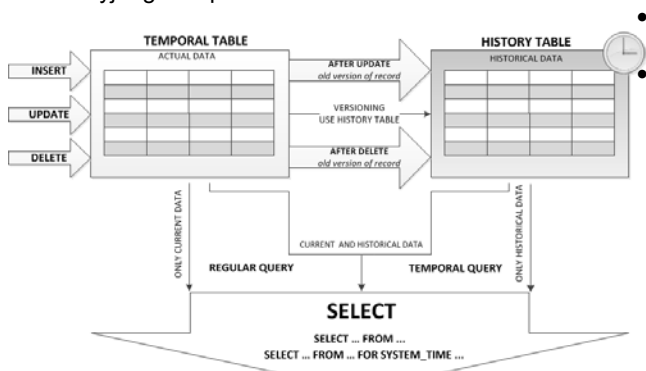
Ponadto tabela temporalna musi zawierać także kolumnę identyfikatora początku transakcji. Stanowi ona trzecią dodatkową kolumnę rozszerzającą definicję tabeli do

postaci tabeli temporalnej w środowisku IBM Db2. Jeżeli w ramach jednej transakcji modyfikowanych lub usuwanych jest wiele rekordów wówczas wartość identyfikatora transakcji jest identyczna dla wszystkich modyfikowanych lub usuwanych rekordów. Wartość tego identyfikatora jest unikatowa w odniesieniu do innych wartości generowanych przez inne transakcje. Tym samym pozwala ona na jednoznaczną identyfikację rekordów temporalnych modyfikowanych przez konkretną transakcję.

Jeżeli rekord jest modyfikowany wielokrotnie w ramach pojedynczej transakcji, w tabeli historii utrwalana jest jego ostatnia (najnowsza) wersja historyczna sprzed modyfikacji. Jeżeli rekord zostanie wstępnie dodany do tabeli temporalnej wersjonowanej systemowo, w ramach transakcji, która zostanie wycofana wówczas taki rekord nie zostanie odnotowany w tabeli historii.

Tabela historii posiada identyczną strukturę jak tabela temporalna z czasem systemowym. Kolumny w obu tabelach muszą mieć zdefiniowane: takie same nazwy, tą samą kolejność kolumn oraz takie same typy danych.

Na rysunku 1 przedstawiono mechanizm składowania i przetwarzania danych temporalnych z obsługą czasu transakcyjnego na platformie IBM Db2.



Rys.1. Mechanizm składowania danych temporalnych z obsługą czasu transakcyjnego na platformie IBM Db2, źródło: opracowanie własne

Tabela temporalna wersjonowana systemowo może pełnić rolę obiektu nadrzędnego lub podrzędnego z punktu widzenia ograniczeń referencyjnych. Jednakże nie dotyczy to tabeli historii. Ograniczenia referencyjne stosowane są tylko w odniesieniu do danych bieżących. W przypadku, gdy tabela temporalna wersjonowana systemowo jest obiektem podrzędnym, w celu wyeliminowania niespójności należy zadbać o to, aby tabela nadrzędna także była tabelą wersjonowaną systemowo.

Warto także wspomnieć o ograniczeniach dla tabeli historii, jakie ona musi spełniać, to m.in.: brak możliwości jawnego jej usunięcia (usuwana jest niejawnie w momencie usunięcia tabeli temporalnej wersjonowanej systemowo, z którą jest powiązana), z punktu widzenia ograniczeń referencyjnych nie może ona występować w roli tabeli nadrzędnej, podrzędnej ani odwołującej się sama do siebie. Nie można także jawnie zmieniać schematu tabeli historycznej tj. dodawać, usuwać ani zmieniać pól tej tabeli.

Tworzenie tabeli temporalnej z obsługą czasu transakcyjnego w IBM Db2

Tabela temporalna wersjonowana systemowo musi mieć zdefiniowane trzy kolumny typu **TIMESTAMP** określające odpowiednio początek i koniec przedziału czasowego oraz czas rozpoczęcia transakcji modyfikującej dany rekord temporalny, który określa, kiedy dana transakcja miała wpływ na dane (zmodyfikowała dane).

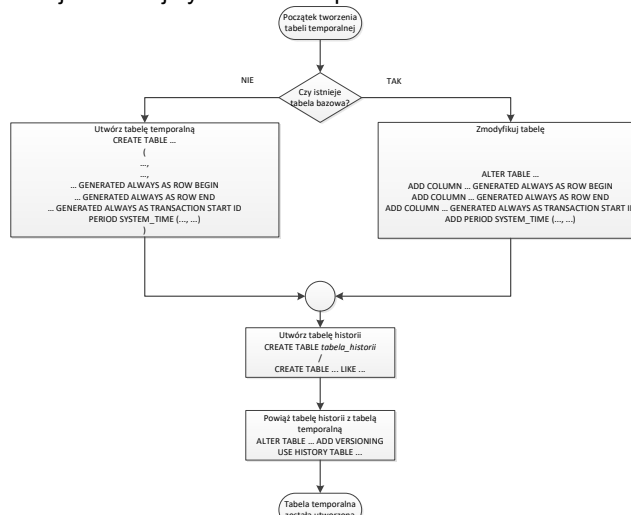
W tabeli 1 zestawiono wykaz systemowych typów danych kategorii data i czas dostępnych na platformie IBM Db2 wraz z zakresem składowanych wartości [16].

Tabela 1. Systemowe typy danych kategorii data i czas dostępne na platformie IBM Db2

Typ danych	Zakres	Dokładność
Date	0001.01.01 +9999.12.31	1 dzień
Time	00:00:00+24:00:00	1 sekunda
Timestamp	0001.01.01 +9999.12.31	0s+1ns
Timestamp without time zone	0001.01.01 00:00:00.000000000 +9999.12.31 24:00:00.000000000	0s+1ns
Timestamp with time zone	0001.01.01 +9999.12.31	0s+1ns

Sposoby tworzenia tabeli temporalnej wersjonowanej systemowo w środowisku IBM Db2 [17, 18]:

- tabela temporalna z jawnymi znacznikami czasowymi,
 - tabela temporalna z niejawnymi znacznikami czasowymi,
 - adaptacja istniejącej tabeli bazowej do tabeli temporalnej z jawnymi znacznikami czasowymi,
 - adaptacja istniejącej tabeli bazowej do tabeli temporalnej z niejawnymi znacznikami czasowymi.
- Na rysunku 2 przedstawiono w postaci schematu blokowego algorytm tworzenia tabeli temporalnej wersjonowanej systemowo na platformie IBM Db2.



Rys.2. Algorytm tworzenia tabeli temporalnej z obsługą czasu transakcyjnego na platformie IBM Db2, źródło: opracowanie własne

Poniżej przedstawiono przykład tworzenia tabeli temporalnej wersjonowanej systemowo z jawnymi znacznikami czasowymi:

```
CREATE TABLE pracownik_tt
(
  IdPracownika INT NOT NULL,
  Imie VARCHAR(15) NOT NULL,
  Nazwisko VARCHAR(30) NOT NULL,
  Miejscowosc VARCHAR(50) NOT NULL,
  SysCzasStart TIMESTAMP(12) NOT NULL GENERATED ALWAYS
  AS ROW BEGIN,
  SysCzasKoniec TIMESTAMP(12) NOT NULL GENERATED ALWAYS
  AS ROW END,
  SysTranID TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS
  TRANSACTION START ID,
  PERIOD SYSTEM_TIME (SysCzasStart, SysCzasKoniec)
);
```

Następnie należy utworzyć tabelę historii, w której będą składowane wszystkie poprzednie (już nieaktualne na daną chwilę czasową, a więc historyczne) wersje rekordu temporalnego. Poniżej zamieszczony został przykładowy fragment kodu tworzenia tabeli historii. Tabela taka musi posiadać identyczną strukturę jak tabela temporalna.

```
CREATE TABLE pracownik_tt_hist
(
  IdPracownika INT NOT NULL,
  Imie VARCHAR(15) NOT NULL,
  Nazwisko VARCHAR(30) NOT NULL,
  Miejscowosc VARCHAR(50) NOT NULL,
  SysCzasStart TIMESTAMP(12) NOT NULL,
  SysCzasKoniec TIMESTAMP(12) NOT NULL,
  SysTranID TIMESTAMP(12) NOT NULL,
);
```

Alternatywnie w celu utworzenia tabeli historii można także posłużyć się poleceniem:

```
CREATE TABLE pracownik_tt_hist LIKE pracownik_tt;
```

Następnie należy powiązać tabelę historii z tabelą temporalną, dzięki czemu możliwe będzie wersjonowanie zmian zachodzących w tabeli temporalnej.

```
ALTER TABLE pracownik_tt
ADD VERSIONING USE HISTORY TABLE pracownik_tt_hist;
```

Kolejny przykład dotyczy tworzenia tabeli temporalnej wersjonowanej systemowo z niejawnymi znacznikami czasowymi.

```
CREATE TABLE pracownik_tt
(
  IdPracownika INT NOT NULL,
  Imie VARCHAR(15) NOT NULL,
  Nazwisko VARCHAR(30) NOT NULL,
  Miejscowosc VARCHAR(50) NOT NULL,
  SysCzasStart TIMESTAMP(12) NOT NULL GENERATED ALWAYS
  AS ROW BEGIN IMPLICITLY HIDDEN,
  SysCzasKoniec TIMESTAMP(12) NOT NULL GENERATED ALWAYS
  AS ROW END IMPLICITLY HIDDEN,
  SysTranID TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS
  TRANSACTION START ID IMPLICITLY HIDDEN,
  PERIOD SYSTEM_TIME (SysCzasStart, SysCzasKoniec)
);
```

W tym przypadku również należy utworzyć tabelę historii, która będzie powiązana z tabelą temporalną. Można to zrobić tak samo jak w poprzednim przypadku. Tworzenie tabeli historii z wykorzystaniem opcji LIKE skutkuje dziedziczeniem przez tabelę historii wszystkich niejawnych, ukrytych pól tabeli temporalnej. Tworzenie tabeli historii w konwencjonalny sposób z podaniem pełnej jej definicji wymaga również w tym przypadku jednoznacznego wskazania pól ukrytych za pomocą opcji IMPLICITLY HIDDEN w odniesieniu do pól ukrytych w tabeli temporalnej.

Poniżej przedstawiono przykład przekształcenia już istniejącej tabeli do postaci tabeli temporalnej wersjonowanej systemowo z jawnie definiowanymi znacznikami czasowymi.

```
ALTER TABLE pracownik_tt
ADD COLUMN SysCzasStart TIMESTAMP(12) NOT NULL
GENERATED ALWAYS AS ROW BEGIN;
ALTER TABLE pracownik_tt
ADD COLUMN SysCzasKoniec TIMESTAMP(12) NOT NULL
GENERATED ALWAYS AS ROW END;
ALTER TABLE pracownik_tt
ADD COLUMN SysTranID TIMESTAMP(12) NOT NULL GENERATED
ALWAYS AS TRANSACTION START ID;
ALTER TABLE pracownik_tt
ADD PERIOD SYSTEM_TIME (SysCzasStart, SysCzasKoniec)
```

W przypadku przekształcenia istniejącej tabeli w tabelę wersjonowaną systemowo do tabeli dodana zostanie kolumna przechowująca początek przedziału czasowego z domyślną wartością 0001-01-01-00.00.00.000000000000. Kolumna końca przedziału czasowego będzie miała przypisaną wartość domyślną 9999-12-30-00.00.00.000000000000. W tym przypadku wartość ta jest celowo o jeden dzień wcześniejsza niż maksymalna możliwa sygnatura czasowa, aby uniknąć możliwego przepełnienia do 5-cyfrowego roku podczas konwertowania wartości na inną strefę czasową.

Po zmianie istniejącej tabeli w tabelę temporalną wersjonowaną systemowo należy jak w poprzednich przypadkach utworzyć dla niej tabelę historii, a następnie obie table powiązać ze sobą włączając tym samym wersjonowanie danych w wymiarze czasu transakcyjnego.

Możliwe jest także przekształcenie istniejącej tabeli do postaci tabeli temporalnej wersjonowanej systemowo z niejawnie ukrytymi kolumnami znaczników czasowych z użyciem klauzuli IMPLICITLY HIDDEN w odniesieniu do tych pól. Przykład takiego rozwiązania pokazano poniżej:

```
ALTER TABLE pracownik_tt
ADD COLUMN SysCzasStart TIMESTAMP(12) NOT NULL
GENERATED ALWAYS AS ROW BEGIN IMPLICITLY HIDDEN;
ALTER TABLE pracownik_tt
ADD COLUMN SysCzasKoniec TIMESTAMP(12) NOT NULL
GENERATED ALWAYS AS ROW END IMPLICITLY HIDDEN;
ALTER TABLE pracownik_tt
ADD COLUMN SysTranID TIMESTAMP(12) NOT NULL GENERATED
ALWAYS AS TRANSACTION START ID IMPLICITLY HIDDEN;
ALTER TABLE pracownik_tt
ADD PERIOD SYSTEM_TIME (SysCzasStart, SysCzasKoniec)
```

Domyślnie usunięcie tabeli temporalnej wersjonowanej systemowo skutkuje także automatycznym usunięciem powiązanej z nią tabeli historii oraz wszystkich indeksów utworzonych w tabeli historii. Opcja RESTRICT ON DROP chroni tabelę historii przed utratą danych historycznych w momencie usuwania tabeli temporalnej powiązanej z tabelą historii. Poniżej przedstawiono przykład utworzenia tabeli historii z użyciem opcji RESTRICT ON DROP:

```
CREATE TABLE pracownik_tt_hist LIKE pracownik_tt WITH
RESTRICT ON DROP;
```

Podsumowanie

Artykuł ten poświęcony został możliwościom składowania danych temporalnych w wymiarze czasu transakcyjnego, w środowisku IBM Db2. Platforma IBM Db2 pozwala tworzyć table temporalne wersjonowane systemowo bezpośrednio na etapie tworzenia takiej tabeli lub poprzez modyfikację schematu istniejącej tabeli. W obu przypadkach klasyczna postać tabeli jest rozszerzana o trzy dodatkowe kolumny przechowujące odpowiednio: początek i koniec przedziału czasowego, w którym dany rekord był rekordem aktualnym oraz początek rozpoczęcia transakcji modyfikującej dany rekord. Użytkownik nie ma możliwości ingerowania w wartości składowane w tych kolumnach, są one automatycznie generowane i utrzymywane przez system. Bieżące wersje rekordów temporalnych przechowywane są w tabeli temporalnej, zaś przeszłe ich wersje składowane są w osobnej tabeli historii powiązanej z tabelą temporalną. Możliwe jest tworzenie tabel temporalnych zawierających jawne i niejawne znaczniki czasowe.

Jeżeli chodzi o zgodność z wymaganiami zapisanymi w standardzie SQL:2011 odnośnie składowania danych temporalnych dla wymiaru czasu transakcyjnego środowisko IBM Db2 zapewnia możliwość tworzenia i obsługi tabel temporalnych wersjonowanych systemowo.

Tabele takie mogą być tworzone zarówno z jawnymi i niejawnymi znacznikami czasowymi [17,18,19,20]. Obsługiwana jest większość typów danych uwzględnionych w zapisach standardu SQL:2011 istotnych dla przetwarzania danych temporalnych oprócz typu INTERVAL. W oparciu o kolumny przechowujące początek i koniec przedziału czasowego tworzona jest definicja czasookresu. Ponadto w środowisku IBM Db2 zaimplementowane zostały predykaty temporalne pozwalające na przetwarzanie danych temporalnych. Zostaną one szerzej zaprezentowane w kolejnym artykule poświęconym przetwarzaniu danych temporalnych dla czasu transakcyjnego na platformie IBM Db2.

Domyślnie nie można usuwać zawartości tabeli temporalnej, co jest zgodne z założeniami temporalnej bazy danych jak i samego standardu SQL:2011. Tabela historii jest automatycznie usuwana w momencie usunięcia tabeli temporalnej, z którą jest ona powiązana. Możliwe jest pozostawienie tabeli historii w momencie usuwania tabeli temporalnej o ile ta utworzona została z opcją RESTRICT ON DROP. Przechowywanie wszystkich poprzednich wersji rekordów temporalnych skutkuje dużym przyrostem danych. W uzasadnionych sytuacjach można rozważyć usunięcie części danych historycznych, co pozwoli zmniejszyć zajętość przestrzeni dyskowej. Zazwyczaj takie usuwanie danych historycznych odbywa się według określonych strategii. Działania takie są możliwe do przeprowadzenia pod warunkiem posiadania odpowiednich uprawnień. Podobne rozwiązania pozwalające zarządzać czasem retencji danych dostępne są także na innych platformach bazodanowych tj. MS SQL Server [3,21], ORACLE [4,22], MariaDB [6,7,23]. W celu zwiększenia wydajności przenoszenia poprzednich wersji rekordu do tabeli historii zmian można użyć opcji APPEND ON. Pozwala ona dołączać nowe wpisy na końcu tabeli historii, pomijane jest wyszukiwanie wolnego miejsca w tabeli historii.

Kolejnym etapem pracy będzie zbadanie stopnia realizacji założeń rozszerzeń temporalnych standardu SQL:2011 na platformie IBM Db2 w kontekście przetwarzania danych temporalnych dla wymiaru czasu transakcyjnego oraz prezentacja zaimplementowanych funkcjonalności temporalnych.

Autorzy: dr inż. Sebastian Łacheciński, Uniwersytet Łódzki, Instytut Logistyki i Informatyki, Katedra Informatyki Ekonomicznej i Medycznej, 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] Łacheciński S., Przetwarzanie danych temporalnych dla wymiaru czasu transakcyjnego na platformie MariaDB, *Przegląd Elektrotechniczny*, 98 (2022), nr 11, 307-312
- [8] Łacheciński S., Składowanie danych temporalnych dla wymiaru czasu rzeczywistego na platformie MariaDB, *Przegląd Elektrotechniczny*, 98 (2022), nr 12, 124-128
- [9] Łacheciński S., Przetwarzanie danych temporalnych dla wymiaru czasu rzeczywistego na platformie MariaDB, *Przegląd Elektrotechniczny*, 99 (2023), nr 6/2023, 111-118
- [10] Date C.J., Darwen H., Lorentzos N., *Time and relational theory Temporal Databases in the Relational Model and SQL*, 2014, Morgan Kaufmann
- [11] Kulkarni K., Jan-Eike Michels, 2012, *Temporal features in SQL:2011*, 34-43 <https://sigmodrecord.org/publications/sigmodRecord/1209/pdfs/07.industry.kulkarni.pdf>
- [12] Temporal extension SQL: https://link.springer.com/content/pdf/10.1007%2F978-1-4899-7993-3_80729-1.pdf
- [13] IBM Db2 WhatsNewDB2: https://public.dhe.ibm.com/ps/products/db2/info/vr101/pdf/en_US/DB2WhatsNew-db2q0e1011.pdf?_gl=1*a37ez1*_ga*NTI0OTc3MzcxLjE2ODg5ODYyMzE.*_ga_FYECCCS21D*MTY5MTkyNDQ0NC4yMS4wLjE2OTE5MjQ0NTYuMC4wLjA
- [14] IBM Db2 Temporal Tables: <https://www.ibm.com/docs/en/db2-for-zos/12?topic=tables-temporal-data-versioning>
- [15] IBM Db2 Creating system period table <https://www.ibm.com/docs/en/db2-for-zos/12?topic=tables-creating-system-period-temporal-table>
- [16] IBM Db2 List datetime values <https://www.ibm.com/docs/en/db2/11.5?topic=list-datetime-values>
- [17] IBM Db2 Language creating system period table <https://www.ibm.com/docs/en/i/7.3?topic=language-creating-system-period-temporal-table>
- [18] IBM Db2 sptt creating <https://www.ibm.com/docs/en/db2/10.5?topic=sptt-creating>
- [19] Petković D., Support of Temporal Data in Database Systems, *International Journal of Computer Applications* (0975 –8887), Volume 152 –No.4, October 2016, 26-33
- [20] Petkovic D., Temporal Data in relational Database Systems: A Comparison, Conference: WorldCIST (1) 2016, At Recife, Volume 1
- [21] MS SQL Server Retention: <https://docs.microsoft.com/enus/sql/relational-databases/tables/manage-retention-of-historical-data-in-system-versioned-temporal-tables?view=sqlserver-ver15>
- [22] Database Development Guide, Using Oracle Flashback Technology: https://docs.oracle.com/database/121/ADFNS/adfn_flashback.htm#ADFNS1008
- [23] MariaDB Removing old history: <https://mariadb.com/kb/en/system-versioned-tables/#removing-old-history>