

Partycjonowanie tabel

Czas: 2 × 45 min

praca indywidualna lub w parach

Cel: Podzielenie dużej tabeli pomiarów na partycje metodami RANGE COLUMNS, LIST COLUMNS i HASH, sprawdzenie przycinania partycji (pruning) w EXPLAIN oraz porównanie usuwania starych danych przez DELETE i DROP PARTITION.

Po zajęciach potrafisz:

- tworzy tabelę partycjonowaną RANGE COLUMNS według daty oraz LIST COLUMNS i HASH
- sprawdza w EXPLAIN i information_schema.PARTITIONS, które partycje czyta zapytanie i ile mają wierszy
- porównuje czas usuwania starych danych przez DELETE i DROP PARTITION
- dodaje nowe partycje poleceniem REORGANIZE PARTITION
- wyjaśnia ograniczenia partycjonowania (klucz główny, klucze obce)

Sprzęt i materiały

- komputer z XAMPP (MariaDB 10.4) lub MySQL 8.4
- plik `r2_24_pomiary.sql` z folderu `Pliki_do_zadan`

Oprogramowanie

- **XAMPP (MariaDB, phpMyAdmin)** — konsola: XAMPP Control Panel → Shell, polecenie `mysql -u root`
apacheFriends.org
- **Edytor tekstu (VS Code lub Notatnik++)** — plik `zadanie24.sql` z zapytaniami
code.visualstudio.com

Przygotowanie przed zajęciami

- Przypomnij sobie z lekcji 2.09 polecenie EXPLAIN i znaczenie kolumn type, key i rows.

Bezpieczeństwo

- Pracuj wyłącznie w bazie `pomiary`. Nie zmieniaj bazy `sklep` ani baz innych osób.
- DROP PARTITION i TRUNCATE PARTITION usuwają dane bez możliwości cofnięcia (nie działa ROLLBACK). Przed każdym takim poleceniem sprawdź nazwę tabeli i partycji.
- Import i kopiowanie 500 tys. wierszy obciążają dysk — nie przerywaj importu w połowie; jeśli się nie powiodł, zaimportuj plik ponownie (sam usuwa starą bazę `pomiary`).

Przebieg zadania

1 Import danych i ograniczenia partycjonowania

Baza pomiary ma tabelę czujniki (40 czujników w halach A–D) i tabelę odczyty (temperatura co ok. 47 s od 1 stycznia do 29 września 2026).

1. Zimportuj plik `r2_24_pomiary.sql` (phpMyAdmin → Import albo w konsoli `mysql -u root < r2_24_pomiary.sql` uruchomione w folderze z plikiem). Wszystkie polecenia zapisuj w pliku `zadanie24.sql`.
2. Z1. Napisz zapytanie zwracające liczbę wierszy oraz najwcześniejszą i najpóźniejszą datę w tabeli odczyty.
3. Z2. Odczytaj z `information_schema.TABLES` liczbę wierszy (`TABLE_ROWS`) i rozmiar w MB (`DATA_LENGTH + INDEX_LENGTH`) tabel bazy pomiary.
4. Z3. Spróbuj podzielić istniejącą tabelę odczyty poleceniem `ALTER TABLE odczyty PARTITION BY RANGE COLUMNS (czas) (...)` z dwiema partycjami: do 1 kwietnia 2026 i `MAXVALUE`. Zapisz numer i treść błędu.
5. Z4. Utwórz testową tabelę `t_test` z kolumnami `id` (klucz główny `AUTO_INCREMENT`) i `czas` `DATETIME`, partycjonowaną jak w Z3 (bez klucza obcego). Zapisz numer błędu i wyjaśnij go, a potem popraw klucz główny tak, by tabela powstała.

2 Partycje według zakresu dat (RANGE COLUMNS)

Utwórz kopię tabeli odczyty podzieloną na kwartały. Kolumna `czas` musi należeć do klucza głównego, a tabela partycjonowana nie może mieć kluczy obcych.

```
CREATE TABLE odczyty_p (  
  id          INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  id_czujnika SMALLINT UNSIGNED NOT NULL,  
  hala        VARCHAR(10) NOT NULL,  
  czas        DATETIME NOT NULL,  
  temperatura DECIMAL(5,2) NOT NULL,  
  PRIMARY KEY (id, czas)  
)  
PARTITION BY RANGE COLUMNS (czas) (  
  PARTITION p2026q1 VALUES LESS THAN ('2026-04-01'),  
  -- uzupełnij: p2026q2, p2026q3  
  PARTITION pmax   VALUES LESS THAN (MAXVALUE)  
);
```

1. Z5. Uzupełnij i wykonaj polecenie z ramki, a następnie skopiuj do `odczyty_p` wszystkie wiersze z `odczyty` (`INSERT ... SELECT`). Wykonaj `ANALYZE TABLE odczyty_p`.
2. Z6. Wyświetl z `information_schema.PARTITIONS` nazwę każdej partycji, jej granicę (`PARTITION_DESCRIPTION`) i szacowaną liczbę wierszy, w kolejności partycji.
3. Z7. Policz odczyty z sierpnia 2026 w obu tabelach. Sprawdź plan obu zapytań: w MariaDB `EXPLAIN PARTITIONS SELECT ...`, w MySQL 8.4 zwykłe `EXPLAIN SELECT ...` (kolumna `partitions` jest w nim zawsze). Zapisz kolumny `partitions` i `rows`.
4. Z8. Sprawdź w `EXPLAIN`, które partycje czyta zapytanie o odczyty z okresu 25 marca – 5 kwietnia 2026, a które zapytanie o średnią temperaturę czujnika nr 7 (bez warunku na dacie).
5. Z9. Policz wiersze tylko w partycji `p2026q1`, używając składni `FROM odczyty_p PARTITION (p2026q1)`.

3 Zarządzanie partycjami: usuwanie i dodawanie

Typowe zadanie administratora: usunąć dane starsze niż pół roku i przygotować miejsce na następny kwartał.

1. Z10. Usuń z tabeli odczyty (bez partycji) wszystkie odczyty z I kwartału poleceniem DELETE. Zanotuj czas wykonania podany przez konsolę lub phpMyAdmin.
2. Z11. Usuń te same dane z odczyty_p poleceniem `ALTER TABLE ... DROP PARTITION`. Zanotuj czas i porównaj liczbę wierszy w obu tabelach.
3. Z12. Wstaw do odczyty_p jeden odczyt z 15 listopada 2026 i sprawdź (jak w Z6), do której partycji trafił.
4. Z13. Podziel partycję pmax poleceniem `REORGANIZE PARTITION` na p2026q4 (do 1 stycznia 2027) i nową pmax. Sprawdź, w której partycji jest teraz listopadowy odczyt, a potem opróżnij p2026q4 poleceniem `TRUNCATE PARTITION`.

4 Partycje LIST i HASH

Te same dane można dzielić także według listy wartości albo równomiernie według funkcji skrótu.

1. Z14. Utwórz tabelę odczyty_hala o tych samych kolumnach, partycjonowaną `LIST COLUMNS (hala)`: p_produkcja — hala A i B, p_magazyn — hala C, p_biuro — hala D. Dobierz klucz główny. Skopiuj dane z odczyty_p i sprawdź liczbę wierszy w partycjach oraz EXPLAIN zapytania o maksymalną temperaturę w hali C.
2. Z15. Spróbuj wstawić do odczyty_hala odczyt z hali E. Zapisz numer błędu i zaproponuj, jak przyjąć dane z nowej hali.
3. Z16. Utwórz tabelę odczyty_hash partycjonowaną `HASH (id_czujnika) PARTITIONS 4`, skopiuj dane i porównaj liczbę wierszy w partycjach p0–p3 z wynikiem zapytania grupującego według `id_czujnika % 4`. Sprawdź w EXPLAIN, którą partycję czyta zapytanie o czujnik nr 7.

W sprawozdaniu umieść

- ☐ plik `zadanie24.sql` ze wszystkimi poleceniami Z1–Z16 opisanymi komentarzami `-- Z1` itd.
- ☐ zrzuty ekranu wyników Z6, Z7 (oba plany) i Z14
- ☐ tabela czasów DELETE i DROP PARTITION
- ☐ numery i treść błędów z Z3, Z4 i Z15 z wyjaśnieniem
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego kolumna czas musi być częścią klucza głównego tabeli odczyty_p?
2. Na czym polega przycinanie partycji (partition pruning) i kiedy nie działa? Odwołaj się do wyników Z7 i Z8.
3. Dlaczego DROP PARTITION jest znacznie szybsze od DELETE i czego nie da się po nim zrobić?
4. Do jakiej tabeli w bazie sklep partycjonowanie by się nie nadawało i dlaczego? Kiedy lepiej wystarczy zwykły indeks?

Kryteria oceny

Kryterium	Punkty
Z1–Z4: import, rozmiar tabeli, poprawnie opisane błędy ograniczeń	3
Z5–Z9: tabela RANGE COLUMNS, odczyt partycji, poprawnie odczytany pruning w EXPLAIN	6
Z10–Z13: pomiar DELETE vs DROP PARTITION, REORGANIZE i TRUNCATE PARTITION	4
Z14–Z16: tabele LIST COLUMNS i HASH z analizą rozkładu wierszy	4
Odpowiedzi na pytania kontrolne	3
Razem	20

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).