

## Program szkolenia:

# Wysokowydajne MySQL, wprowadzenie do technik optymalizacji zapytań SQL i schematów baz danych

## Informacje:

<b>Nazwa:</b>	<b>Wysokowydajne MySQL, wprowadzenie do technik optymalizacji zapytań SQL i schematów baz danych</b>
<b>Kod:</b>	<b>mysql-performance</b>
<b>Kategoria:</b>	MySQL
<b>Odbiorcy:</b>	developerzy
<b>Czas trwania:</b>	2 dni
<b>Forma:</b>	60% wykłady / 40% warsztaty

Szkolenie zostało przygotowane z myślą o programistach i architektach, wykorzystujących w realizowanych projektach bazę danych MySQL.

Baza danych jest jednym z najczęstszych powodów niskiej wydajności aplikacji webowych. Mało optymalny schemat bazy czy nieoptymalne zapytania SQL automatycznie przekładają się na spadki wydajności kolejnych warstw oprogramowania.

Szkolenie zawiera szereg praktycznych rozwiązań i wskazówek opracowanych w dużych i dojrzałych projektach, od prostych środowisk z replikacją master-master po duże, shardowane konfiguracje master-master-(delayed-slave).

## Zalety szkolenia:

- Kruczki, które dadzą Ci realną przewagę
- Dostęp do wiedzy eksperckiej architekta systemów dużej skali (największy polski portal społecznościowy jak i największy komunikator)

## Szczegółowy program:

### 1. Wprowadzenie

1.1. Historia rozwoju MySQL

1.2. Architektura MySQL

1.3. Silniki składowania danych i ich możliwości

1.3.1. InnoDB

1.3.2. MyISAM

1.3.3. MERGE

1.3.4. MEMORY

1.3.5. FEDERATED

1.3.6. ARCHIVE

1.3.7. CSV

1.3.8. BLACKHOLE

1.3.8.1. Profilowanie aplikacji

1.3.8.2. Profilowanie zapytań SQL

1.3.8.3. Profilowanie serwera MySQL

### 2. Identyfikacja wąskich gardeł bazy danych

2.1. Strategie i taktyki testów wydajnościowych

2.2. Dostępne narzędzia i ich możliwości

2.3. Profilowanie wydajności

### 3. Podstawowa optymalizacja schematu bazy danych

3.1. Właściwy dobór typów danych

3.2. Indeksowanie danych

3.2.1. Strategie indeksowania

3.2.2. Selektywność i rozmiary indeksów

**4. Optymalizacja zapytań SQL**

4.1. Metody wykrywania nieoptymalnych zapytań

4.1.1. Narzędzia wbudowane w serwer

4.1.2. Slow-log

4.1.3. Wykorzystanie narzędzi pakietu Percona Toolkit do analizy zbiorczej

4.1.4. Integracja z zewnętrznymi narzędziami

4.1.4.1. Graphite

4.2. Metody monitoringu stanu serwera

4.2.1. Zmienne systemowe

4.2.2. Wykorzystanie danych z INFORMATION SCHEMA

4.3. Plan wykonania zapytania SQL

4.3.1. Sposób obsługi zapytań SQL przez serwer MySQL

4.3.2. Metody analizy planu wykonania pod kątem wydajności

4.3.3. Podpowiedzi dla optymalizatora zapytań

4.4. Optymalizacja określonych typów zapytań

4.5. Tips'n'Tricks

4.5.1. Klucze klastrujące i możliwość ich wykorzystania

**5. Zaawansowana optymalizacja schematu bazy danych i dostępu do danych**

5.1. Replikacja danych

5.1.1. Topologie replikacji

5.1.2. Problemy związane z replikacją i metody ich rozwiązania

5.2. Normalizacja i denormalizacja

5.2.1. Tabele cache-summary

5.2.2. Sharding

5.2.3. Modyfikacje schematu dużych baz danych i związane z tym problemy

5.3. NoSQL-owy dostęp do danych

5.3.1. Handler-Socket i jego możliwości

## 6. Cache'owanie danych

6.1. Zasada działania i ograniczenia natywnego query-cache

6.2. Zewnętrzny cache danych

6.2.1. Cache'owanie danych na poziomie aplikacji

6.2.2. Narzędzia

6.2.2.1. Memcached

6.2.2.2. Redis

## 7. Rozwiązania alternatywne

7.1. Narzędzia pochodne MySQL

7.1.1. MariaDB

7.1.2. Percona Server

7.1.3. Drizzle

7.2. Rozwiązania NoSQL

7.2.1. Przegląd rozwiązań NoSQL wraz z typowymi obszarami zastosowań