


Oracle versus MySQL

Serwery relacyjnych baz danych w portalu
Wirtualna Polska

Remigiusz Sokołowski
Sławomir Cichy



Nieprzespane noce administratora baz danych czyli wymagania aplikacji WWW stawiane serwerom baz danych.

- SQL - ciekawe funkcje, konstrukcje, strojenie wydajności
- Kod proceduralny - PL/SQL, nowości w MySQL
- Użycie kolejek opartych o bazy danych
- Podsumowanie, czyli co wybrać? Oracle czy MySQL?

Kod SQL

SQL to język relacyjnych baz danych. Niestety, tak jak w życiu, w zależności od tego w jakim kraju jesteś, tak w zależności od tego jakiego serwera użyjesz, takim dialektem musisz gadać.

Wyświetlanie szczegółów

Optymalizator regułowy przechodzi do lamusa – obecnie w relacyjnych bazach danych stosowane są optymalizatory kosztowe.

Wspólne dla MySQL i Oracle

- Proste zapytania
`SELECT emp_name FROM emp WHERE emp_id=1;`
- Optymalizacja poprzez indeksy B-Tree
- Zmienne wiązane

Specyficzne dla Oracle

- Optymalizacja poprzez indeksy Reverse, Bitmapy
- Używanie szerokiej gamy wskazówek (ang. *hints*) dla optymalizatorów
- Stosowanie szablonów zapytań (ang. *outlines*)

Specyficzne dla MySQL

- Optymalizacja klucza głównego w InnoDB
- Używanie wskazówek dla optymalizatorów

Specyficzne dla Oracle

- Technologia Oracle Text - optymalizacja poprzez indeksy domenowe

```
SELECT rowid, tk, title, score(1) scr
FROM search_table
WHERE contains(text, 'Oracle',1) >0
ORDER BY score(1) desc;
```

Specyficzne dla MySQL

- Optymalizacja poprzez indeksy FULLTEXT w MyISAM
MATCH (col1,col2,...) AGAINST (expr [search_modifier])

```
SELECT emp_id
FROM emp
WHERE MATCH(opis) AGAINST('sumienny');
```

MySQL – użycie klauzuli LIMIT

- `SELECT /*! SQL_CALC_FOUND_ROWS */ emp_name FROM emp ORDER BY emp_name LIMIT 1250, 10;`
- `SELECT found_rows();`

Oracle

Oracle nie ma klauzuli LIMIT!!!!

■ Potrójnie zagnieżdżone zapytanie z ROWNUM

- `SELECT r.emp_name FROM (SELECT rownum offset, o.emp_name FROM (SELECT emp_name FROM emp ORDER BY emp_name) o) r WHERE r.offset >= 10 and r.offset < 20`

■ Podwójnie zagnieżdżone zapytania z funkcjami analitycznymi

- `SELECT r.emp_name FROM (SELECT emp_name, row_number() over (ORDER BY emp_name) offset FROM emp) r WHERE r.offset >= 10 and r.offset < 20`
- *count() over() w celu określenia ilości wszystkich wierszy spełniających warunki*

MySQL

■ Użycie klauzuli AUTO_INCREMENT

- `CREATE TABLE emp(
emp_id int PRIMARY KEY AUTO_INCREMENT,
name varchar(32));`
- `INSERT INTO emp(name) VALUES('test');`
- `SELECT last_insert_id();`

■ Użycie tabeli jako sekwencji (www.planetmysql.org)

- `CREATE TABLE sequence (id INT NOT NULL);`
- `INSERT INTO sequence VALUES (0);`
- `UPDATE sequence SET id=last_insert_id(id+1);`
- `SELECT last_insert_id();`

Oracle

■ użycie sekwencji

- `CREATE SEQUENCE emp_seq;`
- `INSERT INTO emp VALUES(emp_seq.nextval,
'test');`

Oracle – funkcje analityczne

- AVG, COUNT, MAX, MIN, SUM, ROW_NUMBER, ...() OVER()
- Zapytania hierarchiczne

Postać (przykład :))

```
SELECT last_name, emp_id, manager_id, LEVEL FROM emp START
WITH emp_id = 100 CONNECT BY PRIOR emp_id = manager_id
ORDER SIBLINGS BY last_name;
```

Last name	Emp_id	Manager_id	Level
Kowalski	100		1
Nowak	102	100	2
Adamczyk	111	102	3
Polanowski	103	100	2
Berndt	121	103	3

Podsumowanie – klauzula ROLLUP

Przychody

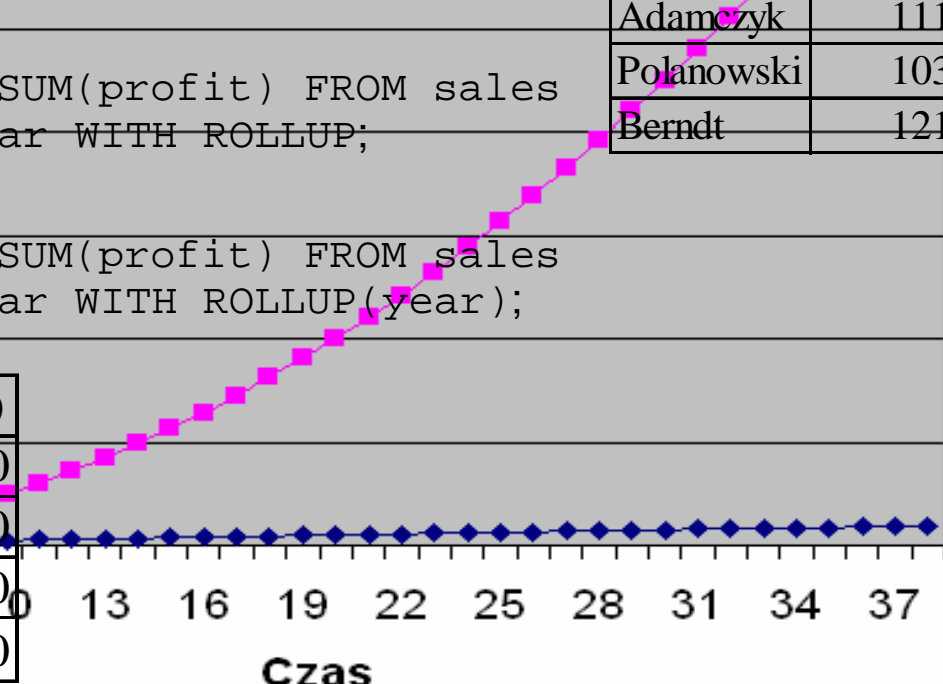
■ MySQL

```
SELECT year, SUM(profit) FROM sales
GROUP BY year WITH ROLLUP;
```

■ Oracle

```
SELECT year, SUM(profit) FROM sales
GROUP BY year WITH ROLLUP(year);
```

Year	SUM(profit)
2004	1000
2005	2000
2006	3000
NULL	6000



◆ Serie 1
■ Serie 2

Obydwa serwery (MySQL od wersji 4.x) wspierają transakcyjne wykonywanie operacji DML

MySQL

- `REPLACE emp SELECT emp_name FROM t_emp;`
- `INSERT INTO emp SELECT emp_name FROM t_emp
ON DUPLICATE KEY UPDATE emp.emp_name =
if(values(emp_name), values(emp_name),
emp.emp_name); (4.1)`

Oracle

- `MERGE INTO bonuses D USING (
SELECT emp_id, salary, dept_id FROM emp
WHERE dept_id = 80) S ON (D.emp_id = S.emp_id)
WHEN MATCHED THEN UPDATE SET D.bonus = D.bonus +
S.salary*.01
WHEN NOT MATCHED THEN INSERT (D.emp_id, D.bonus)
VALUES (S.emp_id, S.salary*0.1);`

Podstawowym narzędziem do optymalizacji wykonania zapytań jest analiza jego planu wykonania.

MySQL

■ SQL

- `DESC SELECT * FROM MYTABLE`
- `EXPLAIN SELECT * FROM MYTABLE`

Oracle

■ Przed analizą planu zapytania musisz się przygotować

- Rola `PLUSTRACE` (`sqlplus\admin\plustrce.sql`)
- Tabela `PLAN_TABLE` (`rdbms\admin\utlxplan.sql`)

■ SQL

- `EXPLAIN PLAN FOR <zapytanie SQL>`
- `SELECT * FROM table(dbms_xplan.display);`

■ SQL*PLUS

- `SET AUTOTRACE ON {EXPLAIN|STATISTICS}`

Oracle ma więcej możliwości poznania co w trawie piszczy. Oprócz ogólnych możliwości poznania planu wykonania zapytania, możemy przeanalizować jego wykonanie za pomocą plików śladu (ang. *trace*).

- ALTER SESSION SET **TRACEFILE_IDENTIFIER**='emp' ;
 - Plik z elementem emp w nazwie w katalogu UDUMP
 - Katalog UDUMP – określony przez USER_DUMP_DEST
- ALTER SESSION SET **SQL_TRACE**=TRUE ;
- ALTER SESSION SET **TIMED_STATISTICS**=TRUE ;
- Plik śladu może być przetworzony narzędziem *tkprof*
 - **tkprof** <plik śladu> <plik wyjścia>
waits=yes sys=yes
sort=fchela,exeela,prsela

Zdarzenie 10046 w Oracle

Podobne do sesji z parametrem `SQL_TRACE=TRUE`

- Plik śladu może być przetworzony narzędziem *tkprof*
- Plik śladu w katalogu UDUMP

Poziom szczegółowości za pomocą parametru `LEVEL`

- 0 – wyłączone
- 1 – identyczne z `SQL_TRACE=TRUE`
- 4 – poziom 1 i zmienne wiązane
- 8 – poziom 1 i zdarzenia
- 12 – poziom 1 i zdarzenia wraz ze zmiennymi wiązanyymi

SQL

```
ALTER SESSION SET EVENTS
  '10046 trace name context forever, level 8';
ALTER SESSION SET EVENTS
  '10046 trace name context off';
```

Kod proceduralny

Jak zrobić auto-numer dla identyfikatora PK w Oracle, który będzie analogiczny do `AUTO_INCREMENT` w MySQL-u? Bez procedur składowanych takich jak wyzwalacze nie pojedziesz...

W zależności od serwera bazy danych mamy dostępne różne jednostki programowe:

Serwer	Oracle	MySQL*
Pakiety	X	
Procedury	X	X
Funkcje	X	X
Wyzwalacze	X	X
Bloki anonimowe	X	X
Typy	X	X

* Nie we wszystkich wersjach MySQL, od 4.1.x wzwyż

- Podniesienie wydajności operacji przetwarzania danych poprzez minimalizację komunikacji pomiędzy klientem a serwerem – serwer zwraca przetworzony wynik
- Uproszczenie aplikacji poprzez przeniesienie części logiki na stronę bazy danych
- Dodatkowe narzędzie kontroli spójności i bezpieczeństwa danych w bazie danych
- Elastyczność – dynamiczny SQL

Wybrane elementy języka Oracle PL/SQL (ang. *Procedural Language/SQL*)

■ DML z klauzulą RETURNING

```
INSERT INTO emp(id, emp_name)
VALUES(emp_seq.nextval, 'Kowalski Tomasz')
RETURNING id INTO :v_id;
```

■ Polecenie EXECUTE IMMEDIATE - natywny dynamiczny SQL

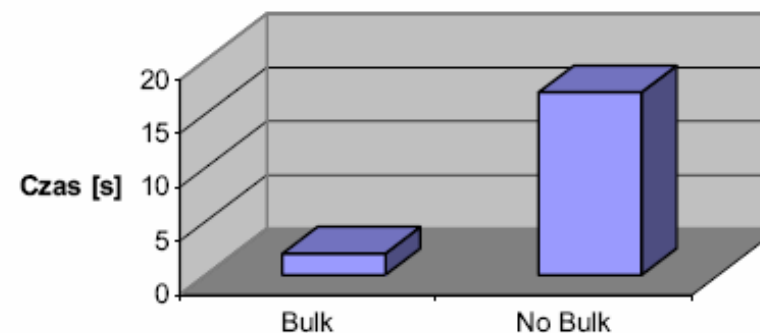
```
v_stmt_str :=
'INSERT INTO dept
VALUES (dept_seq.nextval, :g_dname, :g_loc)
RETURNING dept_no INTO :g_deptno';
EXECUTE IMMEDIATE v_stmt_str
USING v_deptname, v_location, OUT v_deptnumber;
```

■ Dynamiczny SQL przy wykorzystaniu pakietu DBMS_SQL

■ Kolekcje i pętla forall

```
-- deklaracja
TYPE emp_id_t IS TABLE OF emp.emp_id%TYPE;
v_eids emp_id_t;
-- kasowanie danych
FORALL j IN depts.FIRST..depts.LAST
DELETE FROM emp
WHERE department_id = depts(j)
RETURNING employee_id
BULK COLLECT INTO v_eids;
```

Porównanie użycia konstrukcji z BULK i bez BULK



Wybrane elementy języka PL w MySQL-u:

■ Dynamiczny SQL przy wykorzystaniu klauzuli PREPARE i EXECUTE

```
mysql> PREPARE stmt1 FROM 'SELECT SQRT(POW(?,2) + POW(?,2)) AS hypotenuse';
mysql> SET @a = 3; mysql> SET @b = 4;
mysql> EXECUTE stmt1 USING @a, @b;
+-----+
| hypotenuse |
+-----+
| 5 |
+-----+
mysql> DEALLOCATE PREPARE stmt1;
```

■ W klauzuli PREPARE możemy wykonać następujące polecenia: CREATE TABLE, DELETE, DO, INSERT, REPLACE, SELECT, SET, UPDATE oraz SHOW

■ Funkcje i procedury są jak tablice:

```
mysql> CREATE FUNCTION hello (s CHAR(20)) RETURNS CHAR(50)
-> RETURN CONCAT('Hello, ',s,'!');
Query OK, 0 rows affected (0.00 sec)
mysql> SELECT hello('world');
+-----+
| hello('world') |
+-----+
| Hello, world! |
+-----+
1 row in set (0.00 sec)
```

```
mysql> delimiter //
mysql> create procedure read_xa() ↵
  -> begin
  -> select a from x;
  -> end;
  -> //
Query OK, 0 rows affected (0.01 sec) ↵
mysql> delimiter ;
mysql> call read_xa();
+----+
| a |
+----+
| 1 |
| 2 |
+----+
2 rows in set (0.00 sec) ↵
Query OK, 0 rows affected (0.00 sec) ↵
```

Kolejki

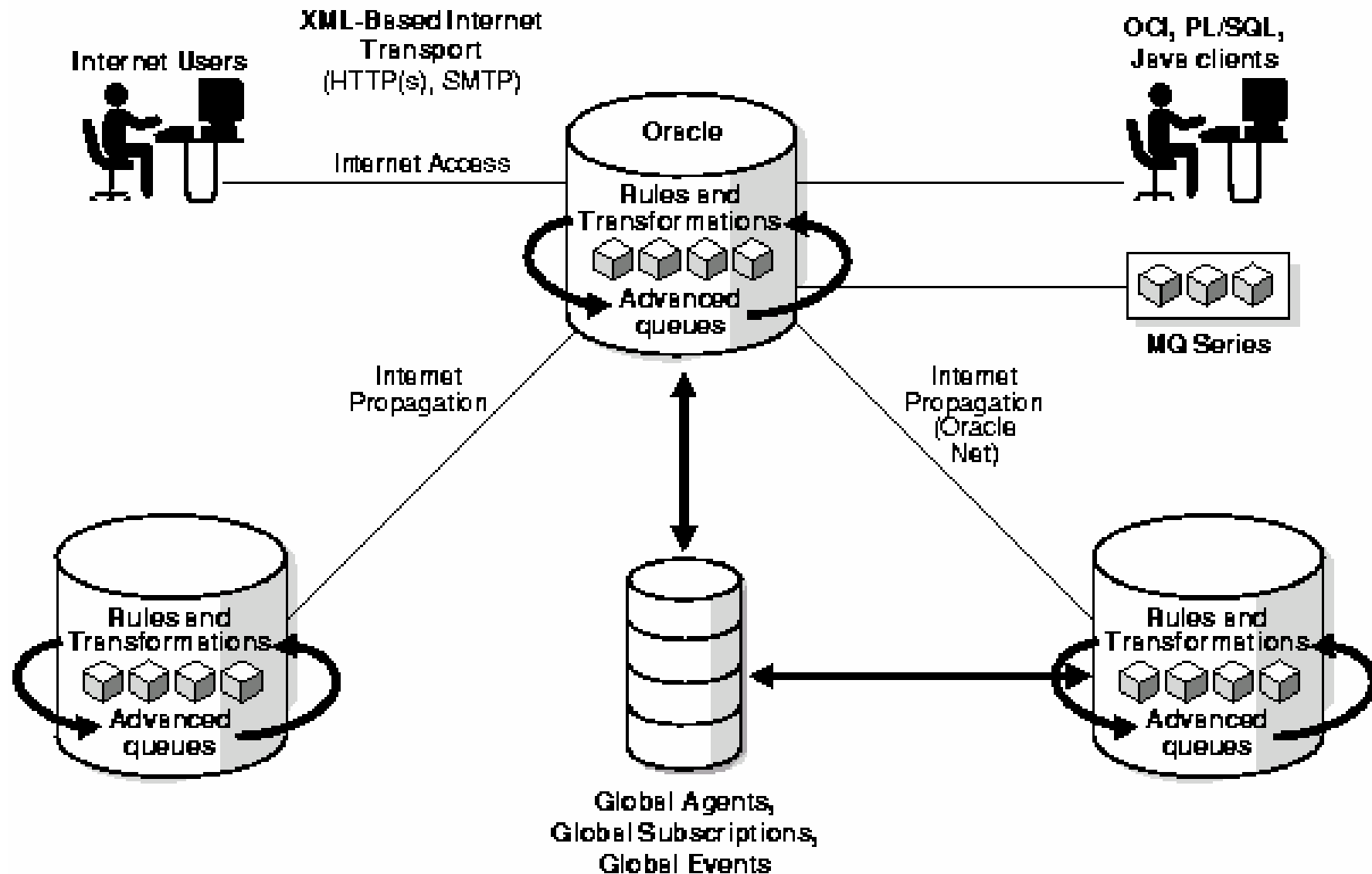
Uwaga! Nadchodzą użytkownicy!



Kolejki – dlaczego?

- Obrona przed konkurencją w dostępie do modyfikacji danych poprzez serializowanie dostępu do nich
- Liczenie statystyk w czasie rzeczywistym?
- Przeniesienie obciążenia modyfikacji danych poza godziny szczytu
- Pomoc w administracji – odłączamy segment, a zmiany na niego czekają
- Asynchroniczna, logiczna replikacja w Oracle
- Współpraca z JMS
- Oracle Advanced Queueing
 - Dojrzała technologia
 - Modele
 - Publish – subscribe
 - Client – server
 - Propagacja do innych kolejek

Kolejki – przykład



Kryteria wyboru

„Być albo nie być – oto jest pytanie...”

„Hamlet” William Shakespeare



Sens użycia nowych technologii

- Primum non nocere

Koszt użycia nowych technologii

- drogie rozwiązania – Oracle
- tanie rozwiązania - MySQL

Internet

- OLTP (MySQL)
- Messaging (Oracle)

Intranet/Extranet

- DSS (Oracle)
- OLTP (Oracle/MySQL)
- Batch load (Oracle/MySQL)
- Messaging (Oracle)

“Kompatybilność” z dotychczasowymi rozwiązaniami

- Know-how posiadanych zasobów ludzkich
- Możliwości techniczne łączenia dotychczas używanych systemów/technologii z nowymi

Przenaszalność pomiędzy architekturami

Posiadane możliwości techniczne (lub finansowe)

- Wymagania baz Oracle
- Wsparcie techniczne

Oba systemy bardzo się różnią

- Oracle udostępnia niektóre bardzo dojrzałe technologie
- Oracle odpowiada na specyficzne potrzeby klientów
- Oracle jest kosztowny
- MySQL obecnie adresuje swój produkt do klientów potrzebujących bazy do aplikacji webowych
- MySQL jest raczej bazą bez dodatkowej zaawansowanej funkcjonalności
- MySQL jest stosunkowo niedrogi

Oba systemy dobrze się uzupełniają

Oficjalna dokumentacja

- dev.mysql.com/doc
- www.oracle.com/technology/documentation/database10gr2.html

Ciekawe zasoby

- www.planetmysql.org
- akstom.oracle.com
- otn.oracle.com
- metalink.oracle.com (dostępny dla klientów)