TEKA Kom. Mot. Energ. Roln. – OL PAN, 2008, 8a, 47–51

METHODS OF DATABASES OPTIMIZATION WITH THE PURPOSE OF THEIR PRODUCTIVITY IMPROVEMENT

[*] Valeriy Dyadychev, [*] Andriy Kolesnikov, [**] Stanisław Sosnowski

[*] The East Ukraine Volodymyr Dahl National University, [**] Uniwersytet Rzeszowski

Summary. The article deals with the optimization of databases. The choice of the theme is determined by the fact that the Internet technologies are widespread and database applications with WEB-interface are widely popular. With the help of normalization, fields indexation, queries sentences making methods usage the task of a particular database productivity improvement in conditions of Web-environment functioning has been solved.

Key words: database, primary key, foreign key, index, indexation, optimization, productivity, normalization, essence, attribute, relations, field, row, query.

INTRODUCTION

This article deals with the question of databases productivity optimization for WEB-environment. Databases have become an essential part of our lives. Databases are commonly used. Applications for databases are the basis for many Internet knots. Besides, applications with WEB-interface are used in corporations networks.

Database is the structured data serving for a definite goal. The interaction with a database is through the application that works with it. The development of Internet technologies has led to a widespread usage of databases. Now they are available not only for big companies. Along with the demand for Internet applications grows the need for databases. There is hardly any WEB-application that is not based on a particular database. Database is a usual tool for a software developer. Databases are specially made for effective organization and search for information.

Keeping information in a database allows either to replace or to add the data storage method in usual files structures. WEB-environment is a very attractive platform for developing and spreading interactive applications oriented on data processing.

In this field the combination of World Wide Web technologies and databases gives new opportunities of creating more and more perfect database applications.

INVESTIGATION OF INDEXATION INFLUENCE ON THE SPEED OF DATA PROCESSING IN DATABASES

With the help of 'EXPLAIN' it can be found out when the tables should be provided with indexes to get a quicker sample that uses indexes for records search.

Let us observe main queries and analyze the purpose of indexes and their influence on the queries performing.

At the stage of logical projecting the indexes of primary key type were set. The analysis of the queries shows that except the key fields the columns ‘name_ahr’ from the table ‘ahr’ and ‘name 3|13|’ from the table ‘enterprise’ and ‘name 3|13|’ from the table ‘concurrent’ are frequently used.

Consequently, the first query is:

It is necessary to choose which of the competitors (concurrents) which works performed for a particular enterprise (“Enterprise1”):

SELECT `concurrent`.`name`, `ahr`.`name_ahr`,
FROM `concurrent`,`concurrent_work`,`enterprise`,`ahr`,
WHERE:
`enterprise`.`name`= “Enterprise1”,
and `concurrent`.`id_concurrent`=`concurrent_work`.`id_concurrent`,
and `enterprise`.`id_enterprise`=`concurrent_work`.`id_enterprise`,
and `ahr`.`id_ahr`=`concurrent_work`.`id_ahr`.
GROUP BY 1

When setting only primary indexes we have the following sample (Fig. 1):

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	concurent	ALL	*NULL*	*NULL*	*NULL*	*NULL*	13	Using temporary; Using filesort
1	SIMPLE	concurent_work	ALL	*NULL*	*NULL*	*NULL*	*NULL*	15	Using where
1	SIMPLE	ahr	eq_ref	PRIMARY	PRIMARY	1	rual myisam no index.concurent_work.id_ahr	1	
1	SIMPLE	enterprise	eq_ref	PRIMARY	PRIMARY	2	rual myisam no index.concurent_work.id_enterprise	1	Using where

Fig. 1. Solving the first query with primary indexes setting

Two tables have been fully scanned, that is proved by the type ‘all’. For obtaining the result we have to make the full scan of both tables, that is on the whole 13*15*1*1 row. We must avoid it. Therefore, let us make the indexation of ‘id_enterprise’, ‘id_concurrent’, ‘id_ahr’ in the table ‘concurrent_work’. The plan of query solving with such indexes is shown in Fig. 2.

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	ahr	ALL	PRIMARY	*NULL*	*NULL*	*NULL*	4	Using temporary Using filesort
1	SIMPLE	concurent_work	ref	id_concurent,id_enterprise,id_ahr	id_ahr	1	rual.ahr.id_ahr	4	Using where
1	SIMPLE	concurent	eq_ref	PRIMARY	PRIMARY	1	rual.concurent_work.id_concurent	1	
1	SIMPLE	enterprise	eq_ref	PRIMARY	PRIMARY	2	rual.concurent_work.id_enterprise	1	Using where

Fig. 2. Solving the first query with extra fields indexes

We can see an improvement: only one table has been fully scanned, in the rest indexes have been used. Only the scan of 4*4 row is performed. If there are several indexes MYSQL selects the most appropriate one. In this case it is the field ‘id_ahr’.

But the used index can not be efficient in all the cases. We can get higher productivity if we place an index on the field ‘name’ in the table ‘Enterprise’ (Fig. 3).

id	select_type	table	type	possible_keys	key	key_len	ref	rows
1	SIMPLE	enterprise	ref	PRIMARY,name	name	23	const	1
1	SIMPLE	concurent_work	ref	id_concurent,id_enterprise,id_ahr	id_enterprise	2	rual.enterprise.id_enterprise	2
1	SIMPLE	concurent	eq_ref	PRIMARY	PRIMARY	1	rual.concurent_work.id_concurent	1
1	SIMPLE	ahr	eq_ref	PRIMARY	PRIMARY	2	rual.concurent_work.id_ahr	1

Fig. 3.

The scan range has been reduced. We need to read only 1*2*1*1 row. Now there is no ‘all’ connection type, ‘ref’ and ‘eq_ref’ types are used.

With the setting of extra indexes we get six times quicker data processing (Table 1) while the data level of ‘concurrent_work’ and ‘enterprise’ tables increases for 32.8% on the whole.

`concurrent_work`.`id_concurrent`, `concurrent_work`.`id_enterprise`, `concurrent_work`.`id_ahr` fields indexation can be useful in other queries as these are the most frequently used fields.

Let us observe and analyze the following query.

Let us analyze the index usage in the query. Let it be necessary to select the pay-rolls on the works (enterprise name, type of works, price, total sum of order) that were carried out in May, 2007.

SELECT `enterprise`.`name`, `ahr`.`name_ahr`, `start`, `price_our`, `sum`,
FROM `enterprise`, `our_work`, `ahr`,
WHERE: `enterprise`.`id_enterprise` = `our_work`.`id_enterprise`,
AND `our_work`.`id_ahr` = `ahr`.`id_ahr`,
AND `start` = ‘2007-03-01’.

The query solving with primary keys only is shown in the fig. 4.

Table 1.

No.	Query Solving Time, Seconds	
	Without Extra Indexes	With Indexes
1.	0.0144	0.0014
2.	0.0235	0.0010
3.	0.0050	0.0016
4.	0.0140	0.0013
5.	0.0046	0.0010
6.	0.0017	0.0014
7.	0.0016	0.0014
8.	0.0460	0.0010
9.	0.0024	0.0014
10.	0.0020	0.0013
11.	0.0052	0.0011
Average Value	0.0080	0.0013

d	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	our_work	ALL	*NULL*	*NULL*	*NULL*	*NULL*	90	Using where
1	SIMPLE	enterprise	eq_ref	PRIMARY	PRIMARY	2	rual.our_work.id_enterprise	1	Using where
1	SIMPLE	ahr	ALL	PRIMARY	*NULL*	*NULL*	*NULL*	3	Using where

Fig. 4.

In this case the first and third tables are fully observed. This leads to 90*1*3=1800 rows scanning.

We may not index ‘id_ah’r and ‘id_enterprise’ in ‘our_work’. Let us index only ‘start’ column. Thus, in all the shown problem situations an index is used. It leads to query solving improvement owing to the reduction of scanned field (fig. 5).

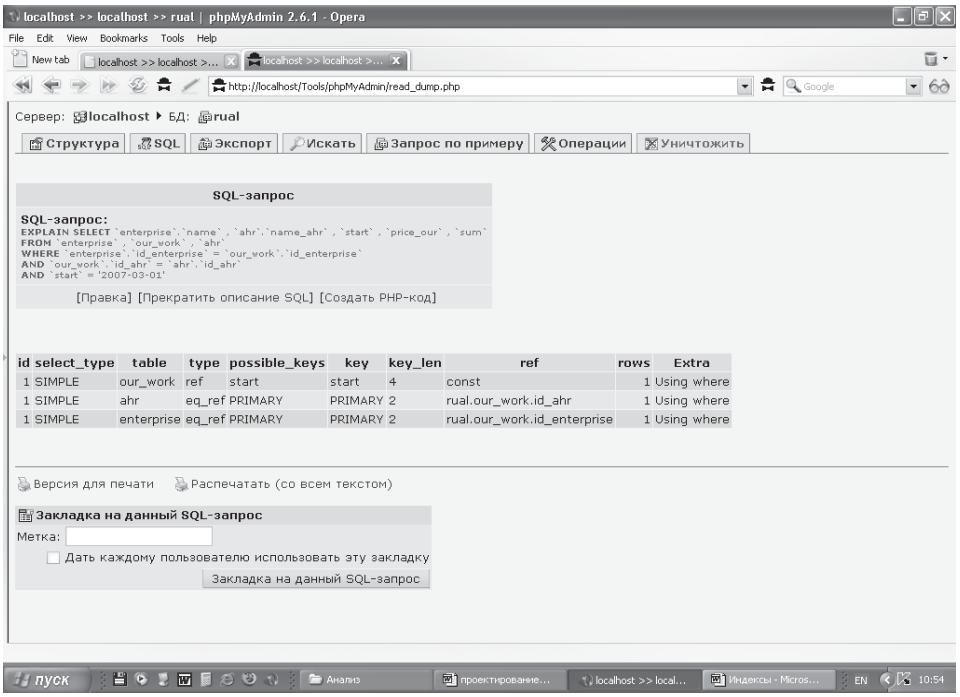

Fig. 5.

CONCLUSIONS

- MYISAM tables type allows to get quicker query solving that is of vital importance for almost any application. The absence of transactions support may be replaced by tables blocking.
- When a statistic tables type is used in a general case we sacrifice the disk space but win in the query solving speed. For most of the table the static type has been selected.
- Indexes setting allows to solve database queries efficiently without extra tables scanning. It helps to reduce the disk reads number and provides high productivity. This peculiarity is especially important if quite a huge data file is accumulated.

- Making correct queries aims to create the conditions for indexes using. It allows to use the advantages of indexation for the sample of the needed data. The opportunity to choose one of the possible variants of query making can optimize the operations of inserting and renewing data. The data extracting query can be solved six times quicker. If the data are inserted the advantage is given to the data inserted from a text file and the operation efficiency grows in 1.9 times. If a huge data file is renewed then it is more useful to postpone the indexes renewing.
- An important peculiarity is the tables operation. Tables optimization with the help of special utilities allows to spare 25% of disc space.
- The setting of server variables allows to improve the queries productivity as well. When not many users are connected and the queries for extraction with data ranking are solved the values are rational. All these settings give an opportunity to solve the query 31% quicker. The correlation of disc reads and read queries equals 0.018. If many queries on data extracting are being solved then it is better to increase the value of reading buffer and keys cache. For quick sorting an important parameter is the size of sorting buffer.

REFERENCES

High Performance MySQL. Optimization, Backups, Replication, Load Balancing & More, 2004. – //O'Reilly & Associates. – 276 p.

Robert L. Bogue "How a relational database solves and optimizes your query". // http: www. techrepublic.com.com/5100-10878_11-5161673.html.

Digo S.M., 2004. – Planning and use of databases: Textbook.-M.: Finances.-208 p.

Frolov A.V., Frolov A.G., 2000. – Databases in the Internet: practical guidance on creation of Web-appendixes with databases. – //M.: Russian Release – 448 p..

Hansen G., 2000. – Databases: development and management. – 704 p.

Battervart P., 1984. – Facilities of defence – in the control systems by databases. Electronics. – -V. 5. – P.46-52.

Deordica Y.S., 1999. – Informative systems and databases. Train aid /. – Lugansk: VUGU, -140 p.

Kirilov V.V. – Bases of planning of databases. // http://www.cs.ifmo.ru.

METODY OPTYMALIZACJI BAZ DANYCH W CELU ULEPSZENIA ICH WYDAJNOŚCI

Streszczenie. Tematem pracy jest optymalizacja baz danych. Wybór tematu podyktowany jest faktem, że technologie internetowe stały się bardzo popularne i aplikacje baz danych szeroko występują w interfejsach WEB. Rozwiązano zadanie znacznego ulepszenia wydajności baz danych w warunkach środowiska WEB poprzez normalizację, indeksację dziedzin oraz formułowanie zapytań.

Słowa kluczowe: baza danych, klucz pierwotny**,** klucz obcy, indeks, indeksacja, optymalizacja, wydajność, normalizacja, esencja, atrybut, relacje, dziedzina, szereg, zapytanie.