

Effects of the Use of Non-Natural Keys on Query Performance in Relational Databases

Mitko Radoev*

Summary:

Choosing a primary key is of a particular importance in the design of a relational database. What are the possible keys, i.e. which columns or combinations of columns are candidate keys? Which of these candidate keys is best suited for a primary key? For quite some time, a trend is gaining more popularity for choosing a primary key - not to use the existing columns of the relationship but to create new, additional column to ensure the uniqueness and to act as a primary key. Recently it became a common practice to use artificial or surrogate identifiers, even in cases where the relationships already have perfectly suited natural identifiers. Using artificial keys is motivated mainly by practical considerations - the keys are shorter, do not have cascade update when changing the key, queries run faster etc. Does using of non-natural keys really increase the performance of the database? This study is testing what is happening with the database in adopting one or the other approach.

Key words: primary key, natural key, non-natural key, query performance

JEL Classification: C 80

1. Introduction

Over the years, almost from the inception of the relational data model, there has been a discussion about the possibilities of using artificial keys and their advantages and disadvantages.

Opinions are still divided on this issue. Even the world-recognized authorities in the field of databases have opposing views.

Recently it became a common practice to use artificial or surrogate identifiers such as:

- AutoNumber,
- Identity,
- Serial,
- Sequence, etc.

1.1. Definitions

Different authors use different terms such as: natural key, relational key, user-controlled key, artificial key, surrogate key, exposed physical locator etc. For the purposes of this study will distinguish two main categories of keys:

Natural key: A key composed of the attributes of the relationship, each of which has a value, which is defined outside the database. Otherwise, we will call natural these keys, which are based on the attributes of the relationship existing in the real world. It

* Senior Assistant Professor, Department of Information Technologies and Communications, University of World and National Economy; e-mail: radoev@abv.bg

does not matter whether these attributes are physical characteristics or other intrinsic properties of objects or are given from state or other authority.

Non-natural key: A key that was created and exists only in the database, which has no analogue in the real world. Otherwise, a non-natural key will understand each key that is not natural, including artificial keys, surrogate keys and physical locators.

1.2. Related Work

The idea for using surrogate keys was set for a first time in 1976 by Hall, Owlett and Todd (Relations and Entities, 1976).

In 1979, during the presentation of the RM/T model Code adopted the idea and proposed the use of surrogate keys (surrogates). These keys are created by the DBMS and are unique to the entire database. Users will not be able to control the value of surrogate keys and this value will not be visible to them. Code, however, specified that introduction of the surrogates does not make user-controlled keys obsolete. Users will often need entity identifiers (such as part serial numbers) that are totally under their control, although they are no longer compelled to invent a user-controlled key if they do not wish to (Codd, 1979).

Most works in this field of research do not contain experimental data. In recent years,

several experimental studies were presented, such as (Link, 2010) and (Slavica Aleksic, 2010); however they are limited in scope.

1.3. Objectives

The main objective pursued by the present study is to determine the impact of the choice of a primary key on the time of execution of queries.

This research will provide contribution to the following areas:

- Design of a hybrid schema in which some relations have natural and others non-natural keys;
- Create multiple queries for data retrieval (Select), and the modification (Insert, Update, Delete), corresponding to the actual queries used in an application.

2. Testing the Query Performance

Tests with changing the data in the main tables are conducted to measure the performance of the database. Queries are performed to insert, update or delete 1000 records in each of these tables.

2.1. Testing Performance of Insert Queries

Table 1 presents changes in the execution times of queries, which are inserting new records in the main tables of the database:

Table 1. Changes in the Execution Times of Insert Queries (%)

Query name	Database with natural keys	Database with non-natural keys	Database with hybrid schema
InsertProperty	100.00%	106.66%	103.33%
InsertTenant	100.00%	118.18%	113.61%
InsertContract	100.00%	97.23%	91.66%
InsertCharge	100.00%	97.50%	97.50%
InsertPayment	100.00%	89.37%	85.10%
InsertPaymentDetail	100.00%	106.91%	110.35%
InsertAccount	100.00%	100.00%	91.42%
Average	100.00%	102.26%	99.00%

2.2. Testing Performance of Update Queries

Table 2 presents changes in the execution times of queries for modification of the contents of the records in the main tables of the database.

Queries which are changing the contents of the primary key are presented separately.

3. Future Work

The study is still ongoing. Further tests must be performed:

- Testing Performance of Select Queries;
- Performing tests among different DBMS;
- Testing with a different number of records;
- Comprehensive evaluation of performance through the use of empirical data on the frequency of execution of queries.

Table 2. Changes in the Execution Times of Update Queries (%)

Query name	Database with natural keys	Database with non-natural keys	Database with hybrid schema
UpdateProperty	100.00%	85.71%	82.15%
UpdatePropertyKey	100.00%	27.16%	27.16%
UpdateTenant	100.00%	81.49%	85.19%
UpdateTenantKey	100.00%	36.99%	87.67%
UpdateContract	100.00%	86.67%	83.32%
UpdateCharge	100.00%	115.17%	112.12%
UpdatePayment	100.00%	89.12%	86.95%
UpdateAccount	100.00%	83.88%	80.64%
<i>Average</i>	<i>100.00%</i>	<i>75.77%</i>	<i>80.65%</i>

2.3. Testing Performance of Delete Queries

Table 3 presents changes in the execution times of queries, which are deleting records from the main tables of the database:

Then the final conclusions will be made about the effect of using different keys on the performance of queries in the database.

Table 3. Changes in the Execution Times of Delete Queries (%)

Query name	Database with natural keys	Database with non-natural keys	Database with hybrid schema
DeleteAccount	100.00%	88.24%	85.28%
DeletePayment	100.00%	94.96%	93.77%
DeleteCharge	100.00%	91.90%	86.49%
DeleteContract	100.00%	100.00%	90.32%
DeleteTenant	100.00%	112.12%	106.07%
DeleteProperty	100.00%	113.88%	105.56%
<i>Average</i>	<i>100.00%</i>	<i>100.18%</i>	<i>94.58%</i>

Articles

References

Codd, Edgar F. 1979. Extending the Database Relational Model to Capture More Meaning. *ACM Transactions on Database Systems*. Dec. 1979, Vol. 4, 4, pp. 397 - 434.

Link, S., Lukovic, I., Mogin, P. 2010. Performance evaluation of natural key and surrogate. Wellington, New Zealand : Technical Report ECSTR10-06, Victoria University, 2010.

Effects of the Use of Non-Natural Keys on Query Performance in Relational Databases

Relations and Entities. Hall P., Owlett, J., Todd, S. 1976. [ed.] G. Nijssen. Amsterdam : North-Holland, 1976. Modelling in Data Base Management Systems. pp. 201-220.

Slavica Aleksic, Milan Celikovic, Sebastian Link, Ivan Lukovic, Pavle Mogin. 2010. Faceoff: Surrogate vs. Natural Keys. *Advances in Databases and Information Systems: 14th East European Conference, ADBIS 2010, Novi Sad, Serbia, September 20-24, 2010, Proceedings*. s.l.: Springer Berlin Heidelberg, 2010.