Database Refactoring to Increase/Retrieve Information From Precision Agriculture Information System

Marcia Beatriz Pereira Domingues, Jorge Rady Almeira Junior,
Wilian Franca Costa, and Antonio Saraiva

Department of Electrical Engineering
Polytechnic School, University of Sao Paulo, Sao Paulo, Brazil

marciabeatrizcp@gmail.com, jorge.almeida@poli.usp.br,
wilianfc@gmail.com, saraiva@usp.br

ABSTRACT

Precision Agriculture (PA) Information Systems improve farm management helping to make the best decisions based on all available information, keeping, controlling, and optimizing resources, returns, and preserving the environment. These systems need to archive, retrieve, and process large amounts of data from many different farms for future analyses. It has to be done without neglecting aspects of entity-relationship model (ER model) at the same time describing all crop cycle's stages. With minor changes in the database schema, the performance of geospatial queries can be improved by refactoring tasks in the literature related to Agile Software Development. A refactoring represents structural, architectural, integrity or data quality change that preserves the system functionalities. As a case study, a PA Information Portal System database that dynamically generates models based on input data was created to represent the relations of input data generated for a Spatial Data Infrastructure used in a PA. In this context, we redesigned this database introducing refactoring techniques in order to improve PA queries performance.

Keywords: Precision Agriculture, Database Refactoring, Evolutionary Databases, Spatial Databases, Query Performance, Brazil.

1. INTRODUCTION

Precision Agriculture (PA) Information Systems improve farm management helping to make the best decisions based on all available information, keeping, controlling, and optimizing resources, returns, and preserving the environment. These systems integrate and process large amounts of data from many different farms (Khosla, 2010). Further complications are the relationship between geo-referenced data and the different ways to process, interpolate and combine them for precision analysis.

The recommended computational solution is the use of spatial databases associated with Web Services to take advantage of the multiple data source providers. A PA

2013 Conference

June 23-27, 2013
TORINO, ITALY

EFITA
WCCA
CIGR

Sustainable Agriculture through ICT innovation

Information Systems needs to archive, retrieve, and process data for future analyses (Santana, 2009). It has to be done without neglecting aspects of entity-relationship model (ER model) at the same time describing all crop cycle's stages. The system needs to resolve geospatial queries to get information back from crop productivity.

With minor changes in the database schema, the performance of geospatial queries can be improved. In the literature related to Agile Software Development these changes are denominated as refactoring tasks (Ambler, 2002). A refactoring represents small structural, architectural, integrity or data quality change, it does not include new functionalities to the system. Ambler wrote a catalog of refactoring with 49 types of refactoring divided in 4 types: architectural, structural, data quality and referential integrity (Ambler, 2006). Ambler discusses in each refactoring the motivations, implications and cautions to its operation, showing the steps for the possible changes of the database. These steps were adapted in this paper to make refactoring tasks into an existing database of PA Information Portal System.

As a case study, a PA Information Portal System database that dynamically generates models based on input data was created to represent the relations of input data generated for a Spatial Data Infrastructure used in a PA. This application has as its main characteristic the insertion of thousands of registers for each user interaction, in which each register represents a geo-referenced polygon, point or pixel for the crop field under analysis. This system is the dynamically geoprocessed generated terrain models based on input data and its respective database persistence. In this context, we redesigned this database introducing refactoring techniques in order to improve queries performance.

1.2 Material and Methods

A spatial database of an existing PA Information Portal System was used to test the database refactoring. A PostGIS extension was used to provide spatial objects for the PostgreSQL database in order to storage and query the information about location and mapping. The refactorings called "Introduce Surrogate Key", "Merge Column", and "Introduce Index" were chosen from Ambler Database Refactoring Catalog. In the proposed experiment, the costs and time consumed for the queries were used to compare the performance between the original model and the refactored model. The tests performed included new codes in database function language (plpgsqq) for data insertion, selection, and cascade delete operations of geographic data through geospatial queries.

Paper organization: Section 2 presents the research related to database refactoring, database evolution and limits of the existent solutions. Section 3 is dedicated to show how an existing PA Information Portal System was refactored in order to improve queries performance and time results. Section 4 presents our conclusions and final remarks.

2. REFACTORING AND DATABASE EVOLUTION

The strategy of refactoring defined by Ambler is applied in complex environments, where there are many applications accessing the same database simultaneously. Ambler proposes a lifecycle model for database refactoring that implements a transition period between the beginning and the end of the refactoring (Ambler, 2002). It is considered that during the transition period, all applications that access the database are changing their source code, in order to be adapted to the new physical structure of the database (Domingues et al., 2011), as illustrated in Figure 1.
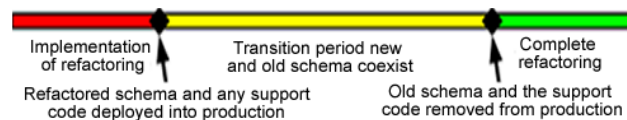

Figure 1. The life cycle of a database refactoring

In a transition period there are the old and the new schemas. Data is stored in both by an asynchronous replication. One or more triggers are implemented to update data. The use of synchronous triggers as support code, proposed by Ambler, present as its limitation the need to avoid possible trigger cycles and deadlocks. To overcome these difficulties the asynchronous approach was proposed by Domingues, Kon and Ferreira (2011).

Asynchronous refactoring process can be done in three steps: collecting, mapping and executing. Those steps perform all refactoring process tasks in which the transactions can be faster than in the synchronous approach and, the trigger cycles can be detected in the execution step. Automated tools can be used to make database refactoring with an asynchronous approach, because there is a single trigger for all refactoring tasks. Both asynchronous and synchronous approaches did not discuss about how many refactoring tasks can be made to perform large changes.

3. CASE STUDY

A spatial database of an existing Precision Agriculture Information Portal System was used to test the database refactoring methods. For this domain, Information Systems can improve farm management, helping to make the best decisions based on all available information, keeping, controlling, and optimizing resources and returns, preserving the natural environment (Khosla, 2010). Management and decision support systems should be designed to meet the specific needs of the farmers (Bongiovanni et al., 2004). Crop analysis is usually performed considering the specific characteristics of each plant population, soil, crop and climate conditions (Zhang et al., 2009).

A PostgreSQL database was used with PostGIS extension to provide spatial data type and functions in order to storage and retrieve the information about location and

mapping. This system aims at meeting the requirements of the productivity data collection and recommending fertilizers application using variable rates (Santana et al., 2007).

The system needs to archive, retrieve, and process data for future analyses without neglecting aspects of entity-relationship model (Chen, 1976). The original PA Information Portal System database model is presented in Figure 2, where there are many relationships that describe the analyzed crop cycle's stages (Murakami et al., 2007).
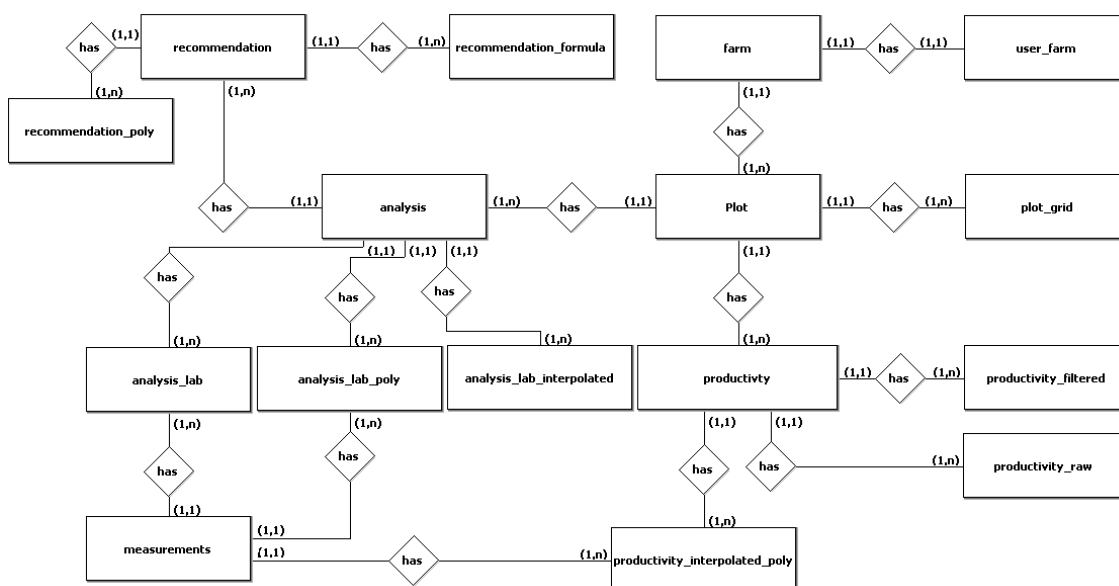

Figure 2. Database original model

The original database model was built with composite natural keys. There is a strong coupling between database model and the PA business domain and some primary key columns such as: farm_id, plot_id, and prod_id that are present in almost all relations. The Figure 3 shows tables farm, plot, productivity and productivity_raw from the original model before refactoring. These tables represent the most important database relations that can be presented as:

- **Farm**: Represents the set of plots used as agricultural crop fields. It can be represented by a multipolygonal geometry (not mandatory).
- **Plot**: A plot contour represents a parcel of the farm that is used as crop field. This entity must have a polygon associated due the fact that this information is used for geoprocessing methods (e. g. machine productivity filtering, interpolation methods, etc.).

2013 Conference
June 23-27, 2013
TORINO, ITALY

Sustainable Agriculture through ICT innovation

EFITA
WCCA
CIGR

- **Productivity**: The productivity represents the harvest for a crop field. This entity represents only the facts (harvesting start date, etc). The real data points are represented in entities **productivity_raw** and **productivity_filtered**.
- **Productivity_raw**: The Productivity_raw entity has the data collected by the harvesting machine as obtained (i.e. without any type of data cleaning).
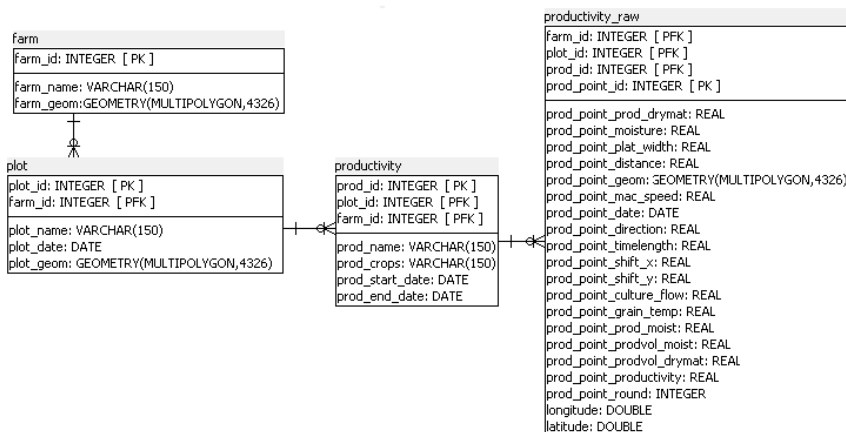

Figure 3. Tables (farm, plot, productivity and productivity_raw) before refactoring

To improve database performance, increase consistency and reduce code complexity all composite natural keys were replaced by surrogate keys in the original model. In order to present the refactoring advantages, tests to insert, select, and delete data were performed in the original and refactored models. Refactoring strategies proposed by Ambler need to be adapted in order to make several changes to the database, such as implementing a common surrogate key strategy across all tables. The process to "Introduce Surrogate Key" refactoring proposed can be performed in a long process in order to implement a set of small refactoring tasks into a refactoring process with changes scripts. This process follows steps below:

- Identify entities to be refactored;
- Identify original primary and foreign key columns;
- Add new ID key column with "serial" datatype;
- Populate ID column with unique values;
- Delete the Original key columns;
- Add foreign column;
- Add referential integrity;
- Write new codes to insert, select and delete operations;
- Test insert, select and delete data;
- Update external access programs.

In the proposed experiment, time consumed for the queries was used to compare the performance between the original model and the refactored model. The tests performed included new codes in database function language (plpgsqq) for data insertion, selection, and propagation of data delete operations of geographic data through geospatial queries. The Figure 4 presents the refactoring proposed to original model, where all composite natural keys were replaced by surrogate keys into the tables chosen for the experiment.
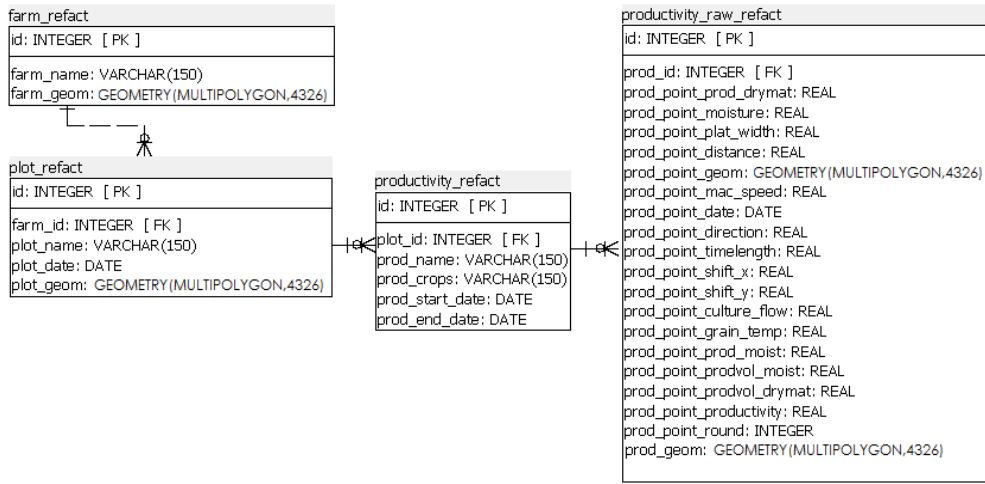

Figure 4. Tables (farm, plot, productivity and productivity_raw) after refactoring

To allow a direct comparison between original and refactored model, the consumed time (ms) was measured. After the refactoring, the time consumed was, 15% shorter to insert, 25% shorter to select, and 4% shorter to delete data for 189.730 rows, from both original and refactored databases. In the Figure 5 we highlight that refactored database performance was better than original considering the time results.
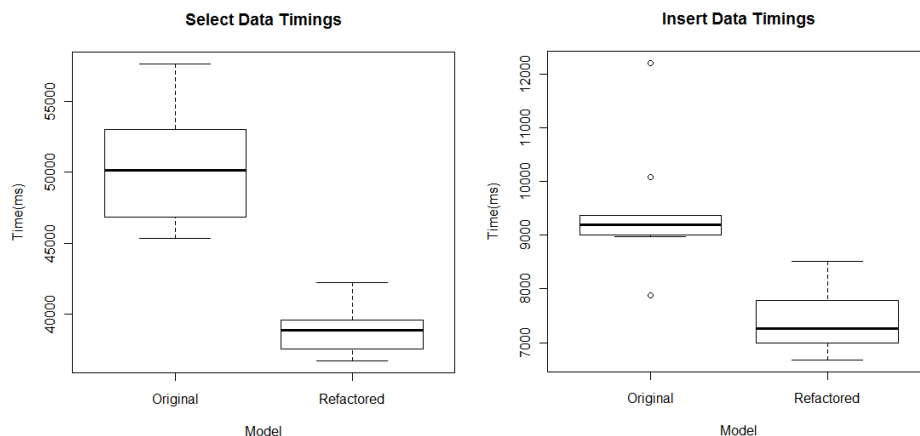
C0194
M. B. P. Domingues, J. R. Almeida Jr., W. F. Costa, A. Saraiva. "Database Refactoring to Increase/Retrieve Information From Precision Agriculture Information System". EFITA-WCCA-CIGR Conference "Sustainable Agriculture through ICT Innovation", Turin, Italy, 24-27 June 2013.

Figure 5. Time results for insert and select data

In the present case study we also performed the "Merge Column" refactoring in order to merge Latitude and Longitude columns of **product_raw** table. To perform this refactoring a new geometry column **prod_geom** was introduced with the previous data from Latitude and Longitude columns. The select operation after this merge was 20% better in terms of time consuming to select data operation than in the original model. The new merged column can be a candidate to be indexed, but when "Introduce Index" refactoring was applied the time consuming to select data increased by 3%. Results for "Merge Column" and "Introduce Index" refactoring were shown in Figure 6.

Figure 6. Time results for select after "Merge Column" and "Introduce Index"

4. CONCLUSIONS

Improving queries performance was the main reason to make changes on the original database. In order to present the **refactoring** advance, all composite natural keys were replaced by surrogate keys into the entities chosen for the experiment. Comparing original and refactored database performance, the scores obtained by the refactoring tasks were higher. For the "Introduce Surrogate Key" refactoring, measuring the consumed time (ms) to insert, delete, and select data was 15%, 4%, 25% shorter respectively. Whereas measurements for queries costs were basically the same to insert and select data, otherwise to delete data operation it was 4% shorter. Aiming at improving the refactored database a "Merge Column" refactoring was made merging "Latitude" and "Longitude" columns, generating an economy of 20% in terms of time consuming to select data operation. The new merged column is a natural candidate to be indexed, but when "Introduce Index" refactoring was applied it resulted in 5% loss from the 20% previously obtained.

C0194
M. B. P. Domingues, J. R. Almeida Jr., W. F. Costa, A. Saraiva. "Database Refactoring to Increase/Retrieve Information From Precision Agriculture Information System". EFITA-WCCA-CIGR Conference "Sustainable Agriculture through ICT Innovation", Turin, Italy, 24-27 June 2013.

5. REFERENCES

Ambler, S. W. 2002. *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process.* New York: John Wiley & Sons, Inc..

Ambler, S. W. and Pramod J. S. 2006. *Refactoring Databases: Evolutionary Database Design.* New York: Addison-Wesley Professional, Inc.

Bongiovanni R. and J. Lowenberg-Deboer. 2004. Precision agriculture and sustainability. *Precision Agriculture* 5(4): 359-387.

Chen, P.P. 1976. The entity-relationship model - toward a uni_ed view of data. *ACMTrans. Database Syst*. 1(1):9-36.

Domingues, H. H., F. Kon and J. E. Ferreira. 2011. Asynchronous replication for evolutionary database development: a design for the experimental assessment of a novel approach. In *Proc. 10th International Conference on Ontologies, DataBases, and Applications of Semantics ODBASE'2011*, 818-825, 18-20 October, Hersonissos, Greece.

Khosla, R. 2010. Precision agriculture: challenges and opportunities in a flat world. In *Proc. 19th World Congress of Soil Science,* 26-29, 1-6 August, Brisbane, Australia

Murakami, E., Antonio M. Saraiva, Luiz C. M. Ribeiro Junior, C. E. Cugnasca, A. R. Hirakawa and P. L. P. Correa. 2007. An infrastructure for the development of distributed service-oriented information systems for precision agriculture. *Journal Computers and Electronics in Agriculture* 58(1): 37-48.

Santana. F.S., I. Y. Gushiken, R. L. Stange, E. Murakami and A. M. Saraiva. 2009. Evolution of a SOA-based architecture for agro-environmental purposes integrating GIS services to an ESB environment In *Proc. 7th EFITA Conference*, 6-8 July, Wageningen, The Netherlands.

Santana, F. S., E. Murakami, A. M. Saraiva and P. L. P. Correa. 2007. A comparative study between precision agriculture and biodiversity modelling information systems. In Proc: *6th EFITA Conference*, 1-6, 2-5 July, Glasgow, United Kingdom.

Zhang, X.; L. Shi, X. Jia, G. Seielstad and C. Helgason. 2009. Zone mapping application for precision-farming: a decision support tool for variable rate application. *Precision Agriculture* 11(2): 103-114.