Journal of Apicultural Research

ISSN: 0021-8839 (Print) 2078-6913 (Online) Journal homepage: https://www.tandfonline.com/loi/tjar20

A note to transfer a generic database pseudocode for storing chronological data from research in apiaries

Fernando Pérez-Rodríguez, Cátia J Neves, Dora Henriques & M Alice Pinto

To cite this article: Fernando Pérez-Rodríguez, Cátia J Neves, Dora Henriques & M Alice Pinto (2019) A note to transfer a generic database pseudocode for storing chronological data from research in apiaries, Journal of Apicultural Research, 58:4, 486-488, DOI: 10.1080/00218839.2018.1517988

To link to this article: https://doi.org/10.1080/00218839.2018.1517988

Published online: 14 Nov 2018.

Submit your article to this journal

Article views: 109

View related articles

View Crossmark data

Journal of Apicultural Research, 2019
Vol. 58, No. 4, 486–488, https://doi.org/10.1080/00218839.2018.1517988

NOTES AND COMMENTS

A note to transfer a generic database pseudocode for storing chronological data from research in apiaries

Fernando Pérez-Rodríguez[a]*, Cátia J Neves[b], Dora Henriques[b] and M Alice Pinto[b]

[a]*Föra forest technologies sll, Av. Eduardo Saavedra, 38, Soria, Spain;* [b]*Mountain Research Centre (CIMO),*
Polytechnic Institute of Bragança, Bragança, Portugal, 5300-253

(*Received 11 May 2017; accepted 30 July 2018*)

In honey bee research conducted in apiaries, a large amount of information is usually generated requiring a flexible database for storing and retrieving data. Here, we developed a generic database pseudocode, based on the abstraction of the apiary system, for data collected from the colonies through time.

Keywords: pseudocode; *Apis mellifera*; colony research; data storage; honey bee system abstraction

Registration and processing of data obtained from colony monitoring and evaluation in a research apiary can be a time-consuming task, depending on the number of colonies and parameters measured, frequency of evaluations, and duration of the experiment (Cauia et al., 2009; Genersch et al., 2010). This task is greatly facilitated by using a database for storing and managing the typically large amount of data collected across the years. The literature reports on a few studies illustrating the utility of having such a database for monitoring colony losses (Mutinelli, 2010) or honey bee enemies (Monceau, Maher, Bonnard, and Thiéry, 2013). Relational database systems are available in different languages and designed for addressing specific goals, like that of Cauia et al. (2009), which was created for honey bee breeding purposes using the proprietary software Microsoft Office^{TM} Access, or that of Zacepins et al. (2016), which implements a cloud database in MySQL for colony monitoring data. There are also proprietary products and applications on cloud for beekeeping management such as that provided by the company Mellarius®. However, there is no freely available database tailored for storing data collected through time for a large number of colony parameters in research apiaries. To fill this gap, the objective of this work was to develop a pseudocode, easily replicable in different database languages, to create a customized database system for storing and managing large amounts of chronological colony data.

The method to develop the logical structure of the pseudocode consisted in the abstraction of the apiary system. The relationships were built using field keys to link and add associated information. The abstraction encompassed the definition and enumeration of all sets and subsets that compose the apiary system, as shown in Figure 1.

The abstraction was based upon a three-level hierarchy, which included the apiary, hive and colony. This approach offered the possibility of adding associated data to each level using the linking fields. At the apiary level, the associated data can be location, climate, flora, ownership, contact data, etc. The database supports an infinite number of apiaries and colonies within the apiary. At the colony level, the associated data can be queen status, colony strength, food reserves (pollen, honey), colony weight, gentleness, swarming, *Varroa destructor* infestation, hygienic behavior, disease symptoms, etc. Finally, at the hive level, the number and type of elements (nest, super, half super) and colony identity can change through time; The number of supers can increase during the honey-flow period and the hive can be occupied by related (daughter queen) or unrelated (replacement of a dead colony by a new swarm) colonies across the years. Therefore, there are no limits in the database for number of apiaries, hives per apiary, hive elements per hive, and colonies that can occupy a single hive through time. The rules used for building the database scheme are specified in Table 1.

The pseudocode was developed to produce a chronological database, so the field "Datetime" appears in each level to register the information associated with the apiary evaluation. This type of architecture allows queries considering the time factor, which greatly facilitates assessment of colony development across the season, for example.

The system considers the queen life cycle allowing one to follow not only the original queen but also its descendants, which will occupy the same hive across the years. To that end, the database includes fields for registering queen's year of birth, status, and different types of queen cells observed during colony evaluation,

*Corresponding author. Email: fernando.perez@fora.es

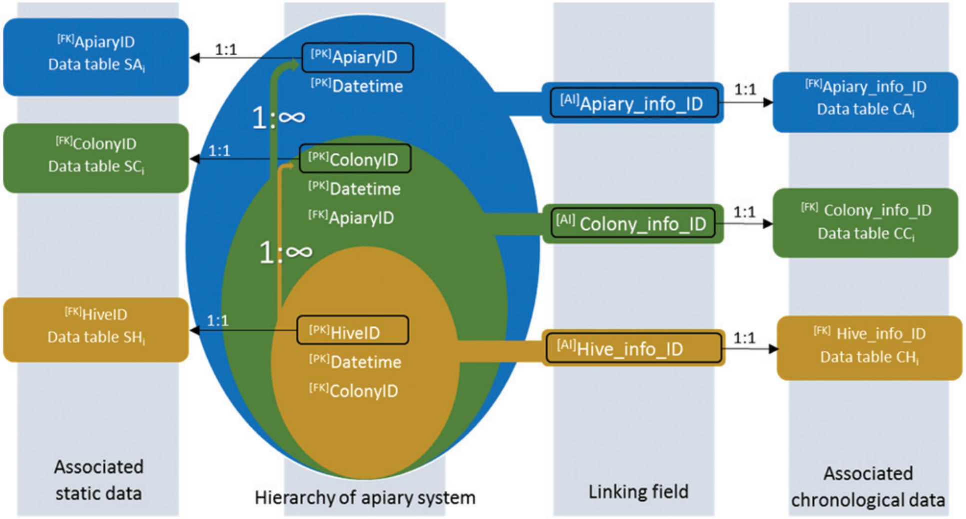

Figure 1. Abstraction of the apiary system with links to add static and chronological data. [PK]: Primary Key, [FK]: Foreign Key, [AI]: Auto Increment, 1:1: relation one-to-one, 1:∞: relation one-to-infinite.

Table 1. Set of Database Relationship Rules (DRR) and Database Operational Rules (DOR) considered in the architecture of the generic database.

Name	Definition
DRR$_1$	One apiary can have an infinite number of colonies
DRR$_2$	One hive can have an infinite number of colonies
DRR$_3$	One colony can be moved between apiaries in different time periods
DRR$_4$	One hive can have an infinite number of elements
DOR$_1$	Every activity (evaluation or management) in the apiary must be registered
DOR$_2$	The honey bee colony has a code that is linked to the natural development of the queen; If a queen swarms, supersedes, or dies and a new queen is produced or is introduced, the colony code is changed

allowing predictions of whether a queen swarmed, was superseded or if an emergency queen was produced. The system also allows one to register movement of colonies (migratory beekeeping) or to follow a colony split, in which case the hive with the old queen keeps the original colony ID, and the queenless half of the colony receives a new colony ID and a new hive box.

The generic database pseudocode was designed to provide a flexible tool adjusted to the complexities involved in colony monitoring and assessment in research apiaries. While the user has a basic structure (apiary, hive, colony), other features can be added any time, depending on the aim and specificity of the research project. The pseudocode allows replication of the database in distinct languages, and can be adapted to create different interfaces and forms to introduce data, minimizing the risks of mistakes, and to make different queries for rapidly generating results, all adapted to the specific requirements of each study or research project. To make the database readily available, we provide an SQL code example in SOURCEFORGE (https://data-bee.sourceforge.io). To our knowledge, this is the first time a pseudocode (and open source code) for building such a flexible database is made freely available for the honey bee research community.

Acknowledgments

We thank J Chávez-Galarza for the fruitful discussions during the design of the database architecture. This research was funded through the 2013-2014 BiodivERsA/FACCE-JPI joint call for research proposals, with the national funders “Fundação para a Ciência e Tecnologia” (Portugal), “Agence Nationale de la Recherche” (France), and “Ministério de Economia y Competividade” (Spain).

ORCID

Fernando Pérez-Rodríguez http://orcid.org/0000-0001-7825-7824

References

Cauia, E., Siceanu, A., Sapcaliu, A., Magdici, M., Chirila, A., & Cauia, D. (2009). The evaluation of honey bee colonies, using a database with application in honey bee breeding programs. Apimondia Congress. Retrieved from https://www.apimondia.com/congresses/2009/Technology-Quality/Symposia/The%20evaluation%20of%20honeybee%20colonies%20by%20using%20a%20data%20base%20software%20with%20application%20in%20honeybee%20breeding%20programs%20-%20CAUIA%20Eliza.pdf

Genersch, E., Von Der Ohe, W., Kaatz, H., Schroeder, A., Otten, C., Büchler, R., Berg, S., Ritter, W., Mühlen, W., Gisder, S., Meixner, M., Liebig, G., & Rosenkranz P. (2010). The German bee monitoring project: A long term study to understand periodically high winter losses of honey bee colonies. *Apidologie*, *41*(3), 332–352. doi:10.1051/apido/2010014

Monceau, K., Maher, N., Bonnard, O., & Thiéry, D. (2013). Predation pressure dynamics study of the recently introduced honey bee killer *Vespa velutina*: learning from the enemy. *Apidologie*, *44*(2), 209–221. doi:10.1007/s13592-012-0172-7

Mutinelli, F. S. (2010). A network for monitoring honey bee mortality and colony losses in Italy as a part of the APENET Research Project. *American Bee Journal*, *150*(4), 39–389.

Zacepins, A., Kviesis, A., Ahrendt, P., Richter, U., Tekin, S., & Durgun, M. (2016). Beekeeping in the future - Smart apiary management. In *Carpathian Control Conference (ICCC), 2016 17th International* (pp. 808-812). IEEE.