# Chapter 48
# Schema Evolution in Conventional and Emerging Databases

**Zouhaier Brahmia**
*University of Sfax, Tunisia*

**Fabio Grandi**
*University of Bologna, Italy*

**Barbara Oliboni**
*University of Verona, Italy*

**Rafik Bouaziz**
*University of Sfax, Tunisia*

## ABSTRACT

*In information systems, changing the database schema is a common but often troublesome task in database administration that is needed for many reasons such as changes in user requirements, compliance to new regulations, addition of new functionalities, or correction of deficiencies in the current schema. For most database applications, changing the schema of the database without loss of existing data is a significant challenge: it is usually a time-consuming and error-prone task which must be done carefully. In the literature, schema evolution has been defined as the modality for the management of schema changes which relieves database programmers and administrators from this burden, by automatically recovering extant data and possibly adapting them to the new schema. The main goal of this chapter is to present the recent research proposals that deal with schema evolution in traditional and emerging databases and to discuss the recent advances on schema evolution support in mainstream DBMSs.*

## INTRODUCTION

In information systems, not only do data change over time, but also database schemata evolve frequently (Sjøberg, 1993). Changing database schema, which is a common but often troublesome task in database administration, occurs for many reasons such as changes in user requirements, compliance to new regu-

DOI: 10.4018/978-1-5225-7598-6.ch048

lations, addition of new functionalities, or correction of deficiencies in the current schema. Usually, the database administrator changes a database schema through a schema definition language (e.g., SQL DDL).

Two fundamental aspects are involved by schema change: (i) *semantics of change*, which deals with the effects of the change on the schema itself, in order to maintain schema consistency after schema changes, and (ii) *change propagation*, which deals with the effects of the change on the underlying data, in order to guarantee data consistency with the modified schema.

For most database applications, changing the schema of the database without loss of existing data is a significant challenge: it is usually a time-consuming and error-prone task which must be done carefully. In the literature (Jensen et al., 1998), *schema evolution* has been defined as the modality for the management of schema changes which relieves database programmers and administrators from this burden, by automatically recovering extant data and possibly adapting them to the new schema. During the last two decades, a lot of theoretical work has been done on schema evolution in conventional databases, within the relational, object-oriented, and XML settings, as described in (Brahmia et al., 2015). In recent years, the schema evolution problem has been addressed also in emerging databases (e.g., temporal, genome, embedded, mobile, and NoSQL databases), as a consequence of the growing use of these new databases by contemporary applications (e.g., digital libraries, mobile applications). However, schema evolution support remains so far almost neglected at the practical level: existing commercial systems (i.e., DBMSs like Oracle and DB2, and schema management tools) only provide a quite limited support. Thus, currently, each database administrator uses ad hoc techniques to manage the evolution of a database schema.

The main goal of this chapter is (i) to present the recent research proposals, not already covered in (Brahmia et al., 2015), that deal with schema evolution in traditional and emerging databases, and (ii) to discuss the recent advances on schema evolution support in mainstream DBMSs. The rest of this chapter is organized as follows. The next section gives some background on our subject. In "Recent Research on Schema Evolution", we present an update on recent research proposals concerning schema evolution. "DBMS Support for Schema Evolution" surveys the support of schema evolution in the state of the art of the latest database technology. Finally, future work directions and conclusion are provided.

## BACKGROUND

In this section, we illustrate with a simple example the functioning of schema evolution, contrasting it with the lowest level of schema change support that can be embedded in a database, that is the modality of schema modification (Jensen et al., 1998). Assume that we have a relational database that contains only an AUTHOR relation with the attributes ID (primary key), NAME, PHONE, and COUNTRY. The first state of this database is shown in Figure 1.

The catalogues store information on the schema S1 of the AUTHOR relation. The table AUTHOR contains two tuples for two authors. Then consider the following schema changes:

*Figure 1.*

**S1**

**AUTHOR(ID, NAME, PHONE, COUNTRY)**

| ID | NAME | PHONE | COUNTRY |
|---|---|---|---|
| 1 | Aicha | 11223344 | Tunisia |
| 2 | Cristiana | 55667788 | Italy |