



---

**DBE Software, Inc.**

6842 Elm Street  
McLean, VA 22101  
USA

E LCS@dbesoftware.com

P +1-703-847-9500

F +1-703-991-2500

## DBE Software and its Value Proposition

DBE Software, Inc. is dedicated to solving the problem of poor information quality due to incorrect design of the database schemas. We have created a software solution that analyzes a database schema, either during creation or after implementation, and diagnoses flaws—relative to the relational theory—that commonly lead to breakdown of information quality.

---

### Quality Information is the Major Corporate Asset

Information is the most important asset of any organization; both government and commercial industry rely on information for (a) daily operations (transactional system) and (b) for decision-making activities (data warehousing and other systems). Without quality information, a company can not operate successfully.

---

### Cost of Poor Data Quality is Huge

According to a recent study by The Gartner Group, the cost of poor data quality to US business is over \$600 billion per year. Another study shows that over 50% of all data warehousing projects fail and, if they succeed, 25% are delivered late and over budget.

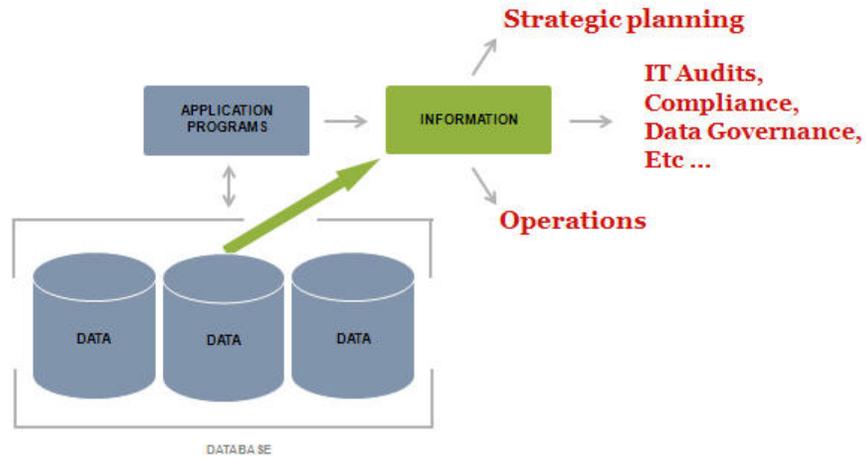
This data quality issue has gained even greater relevance as public companies must now adhere to much closer scrutiny in the form of the Sarbanes-Oxley Act; corporations must now certify that their IT infrastructure relative to the financial reports are without structural flaws and are capable of producing unimpeachable financial reports.

Every corporation is desperately trying to minimize this cost while improving the quality of the information.

---

### How Information is Produced

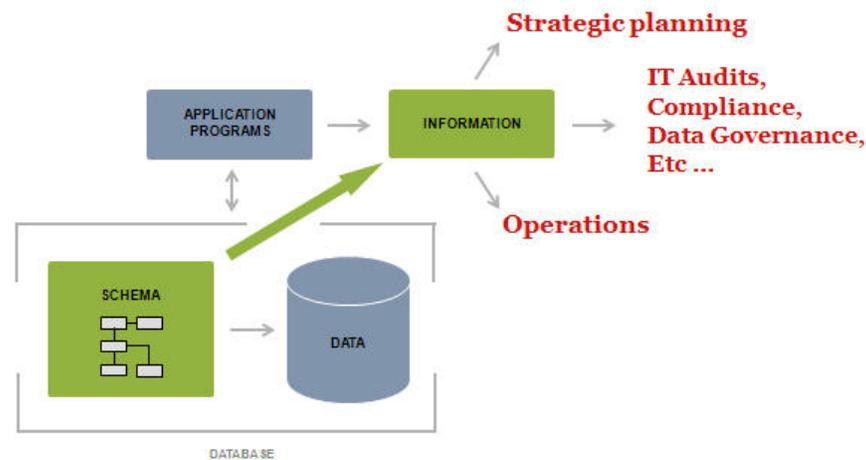
Information is generated by an application program that processes data in a database system. The concept of a database is not properly understood by top management. They normally think of the database as a set of data, which is correct, but incomplete.



**The Short-sighted View of Data Quality**

Because of this perception, Data Quality is seen as a Data Cleansing/Data Profiling activity. Both these procedures are important and must be performed, but they are not nearly enough to ensure the quality of the information.

Even when The Gartner Group talks about Data Quality, they only talk about Data Cleansing/ Data Profiling. Data Quality is only one of the components of Information Quality.



**The Complete Picture, with the Database Schema**

The set of data is managed and controlled by the database schema. The database schema stores all the data definitions, the relationships between data, and the business rules. If the database schema has flaws, the chances of the application producing quality information are very slim.

---

**Quality Information**

By looking at the previous diagram, we can see that the quality of the information produced by the applications depends directly on the quality of:

- The application programs (the code itself);
- The initial data stored in the database and
- The database schema.

---

**Quality of the Programs**

This is an issue that has been studied for a very long time. There are many program development methodologies such as the Software Development Methodologies from the Carnegie Mellon Institute. Most corporations and government entities apply them in one fashion or on another. The quality of the programs has improved substantially in the last several years. This is no longer a major issue.

---

**Quality of the Data Stored in the Database**

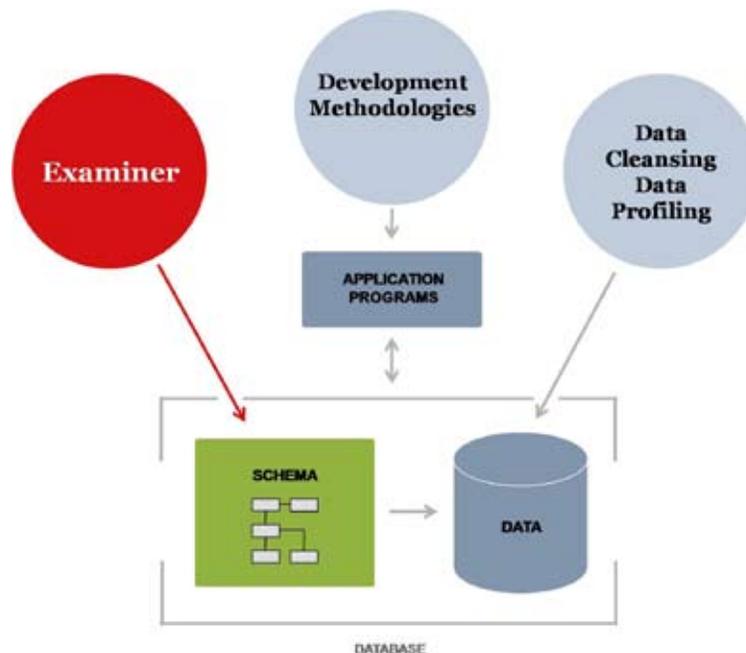
This is where most of the effort is being put to improve data quality. There is a huge industry in the Data Cleansing/Data Profiling space. This effort is extremely important, but not nearly enough.

---

**Quality of the Schema**

Joe Celko ([www.celko.com](http://www.celko.com)), a database guru and published author on relational databases, states that with a poorly designed database schema, it is very difficult to:

- Achieve application performance and
- Produce quality information.



**Modeling Tools are not Enough**

Modeling tools have some minor validation diagnostics, mainly to deal with the completeness of the model, but no integrity and consistency diagnostics. These efficient modeling tools help the user design the data model in a short time, but they do not have the intelligence to debug the data model and, therefore, the database schema.

Companies are making every effort to improve the quality of their data and of their application programs, but very few understand the need for a high quality debugged database schema. They assume that having a good data modeler using a good data modeling tool, such as ERwin®, Sybase PowerDesigner, Oracle’s Designer or Embarcadero®’s ER/Studio®, will ensure their database schemas will be sound and flawless. **This is far from reality.** It is basically impossible for even the best data modeler to completely validate a database schema manually. The schemas are simply too large and complex.

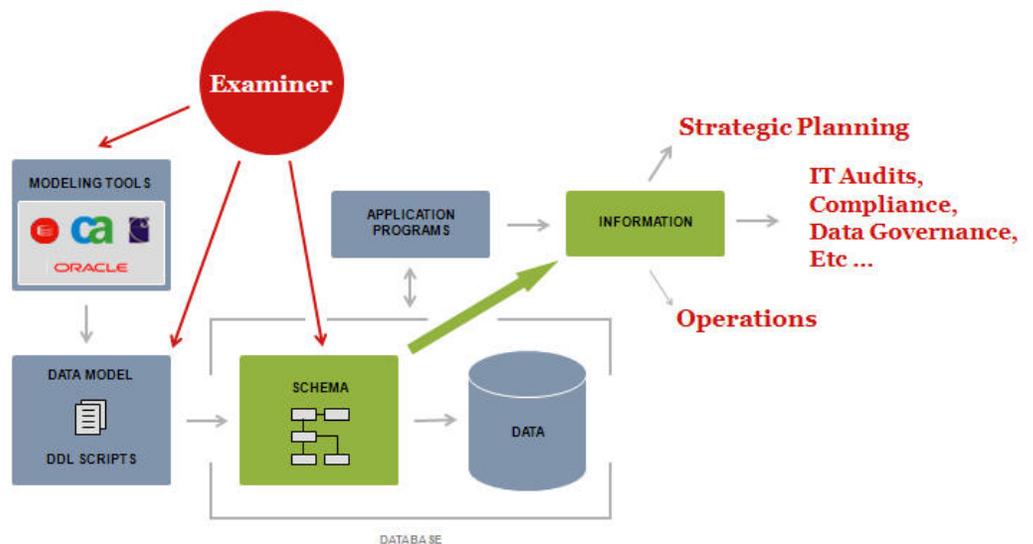
Without a quality schema, the chances of generating quality information are very slim.

**DBE Software’s Database Examiner and the Quality of the Database Schema**

As most of the database schemas are created using modeling tools, Examiner was designed to read the database schema in all its stages – from a data model to a set of DDL scripts to the database schema itself. It provides over 50 diagnostics, explains the issues, make suggestions and, in situations where there is no need for human intervention, generates the corrective DDL scripts.

Examiner uses an extensive knowledge base about the relational model to validate the database schema, DDL scripts or data model; it detects the problems, explains the issues and the impact; makes suggestions on how to fix the issues and, when there is no need for human intervention, generates the corrective DDL scripts.

In summary, Examiner ensures the quality of the database schema, thus improving the quality of the information produced by the application programs.

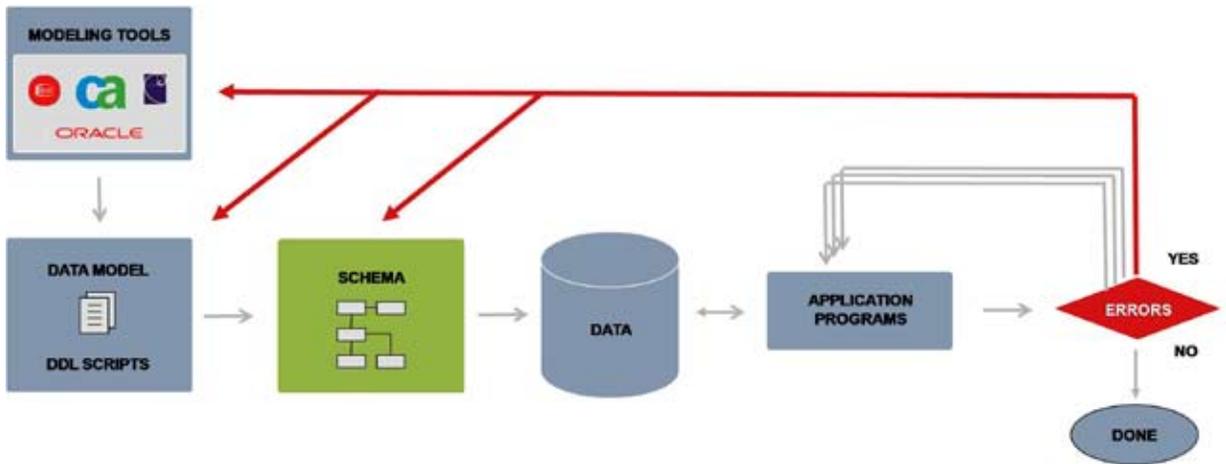


**The Problem with Today's Development Process**

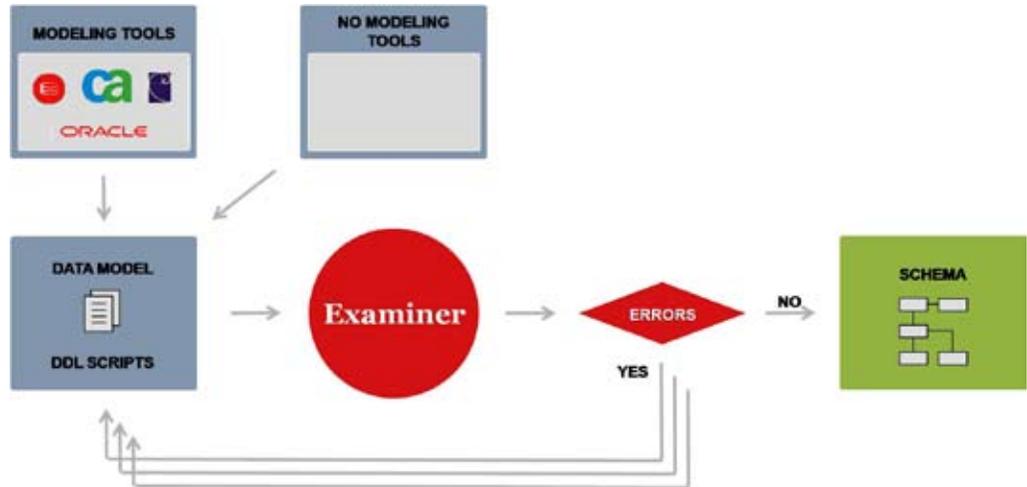
In a standard development process without Examiner, the data model is created manually or using a modeling tool. After some checking of the data model, the database schema is generated and application programs are created. The system is tested as the database is being populated with data. When an error in the code is identified, it is fixed and the testing continues. But if a problem is found with the database schema, it is the data model that must be fixed and the changes reproduced in a database that now has several megabytes of data and many programs that access this data. This is where the costs go up tremendously and the delivery time expands proportionally. In this situation, to fix the schema, you may have to download part of the data, fix the schema, upload the data to a new location on the database. And then evaluate the programs being affected, fix them and re-test them. All this because the debugging of the schema is done at a very late stage in the development process.

**Using Examiner in the Development Process**

Let us now look at the same project using Examiner to debug the database schema at a very early stage of the project, when the schema has not yet been created.



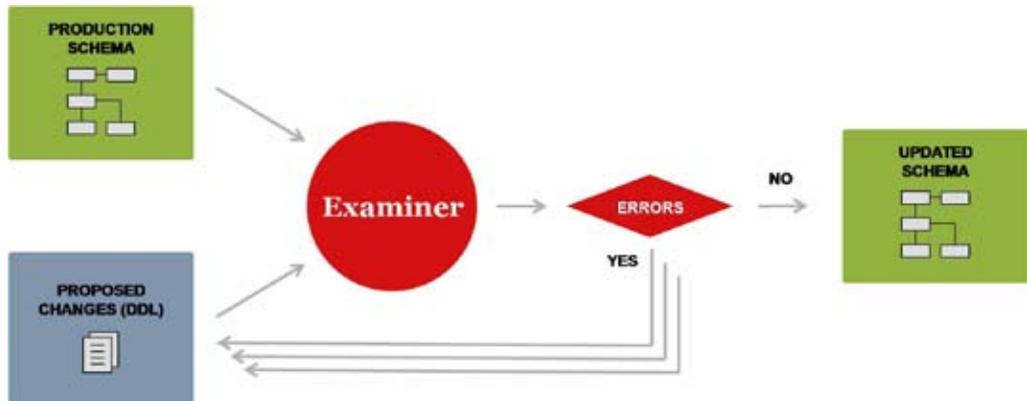
Here we are validating the data model and checking for errors and inconsistencies until they are nonexistent. If we find any problem, we go back to the model and fix it. When there are no errors, we create the database schema. This methodology saves significant time and greatly reduces costs over the trial and error method.



**Use of Examiner in the Maintenance Process**

Another important use of Examiner is when an application needs some improvement that requires changes to the database schema. The ideal situation is to be able to simulate the changes before making them permanent. Examiner allows this by “merging” the production database schema (or a copy for safety reasons) with the proposed DDL scripts for the change. In this case we test the combined “schema” and check for any errors or inconsistencies. If there are problems, we repeat this process until there are no more errors at which time, the updated schema will be created.

Simulating the changes has tremendous advantages over the antiquated process of implementing changes and then looking for the errors.



---

### Other Uses of Examiner

Examiner can and should be used in several instances, such as:

**Data Governance.** Examiner used as a quality control discipline for assessing, managing, using, improving, monitoring, maintaining, and protecting organizational information.

**Sarbanes-Oxley Compliance.** Sections 302 & 404 require that the IT infrastructure relative to the financial reports are adequate; Examiner checks the quality of the database schemas.

**IT Audits.** Today, most of the IT audits only report the quality of the documentation of the programs; by using Examiner, the auditors can do a much better job in their audits.

**Data Warehouse projects.** DBMS Inc, a consulting company from Seattle, used Examiner to check the disparities among 4 different data sources, and after making the recommended changes, the project was selected as the Best Data Warehouse Project of the year, in the Telecommunication category by The Data Warehouse Institute.

**Database Assessment.** Consultants, before bidding for a project, can effortlessly evaluate the work required to make improvement on any application by examining the database schemas and checking on their health.

Examiner is a must-have tool to be used by any organization, private or public that develops corporate applications using relational databases.

Examiner provides the following benefits:

- Modelers will produce better data models/schemas
- Applications will be deployed earlier
- Development costs will be reduced
- Programs will run faster
- Data models/schemas will be evaluated by a standard criteria

It is extremely difficult to measure the ROI of projects that improve the quality of the information, but it is clear that decisions based on quality information provide enormous saving.

A tangible ROI can be measured based on:

- Reduced development time
- Associated reduced development costs
- Reduced delivery time
- Reduced number of applications being scrapped

A couple of ROI examples can be found on our web site.