The Impact of Data Warehouse Automation

Published 30 December 2021 - ID G00760886 - 9 min read

By Analyst(s): Henry Cook

Initiatives: Data Management Solutions

Data warehouses provide organizations with reliable, high-performance analytics, but implementing and maintaining these systems can be daunting. Data warehouse automation tools are maturing and can provide data and analytics leaders greater productivity and agility in developing these systems.

Overview

Impacts

For data and analytics leaders, data warehouse automation (DWA):

- Provides a new style of development that all data warehouse development organizations should consider
- Makes designers, developers and users more productive
- Provides greater agility, speeding the rate of change within data warehouse systems

Recommendations

Data and analytics leaders responsible for modernizing data warehouses or data lakes within their data management solutions landscape should:

- Adopt the use of DWA if supported by a productivity-based business case.
- Assess whether all or part of their data warehouse development can benefit from DWA by identifying processes that can be automated for quicker deployment.
- Adopt DWA tools to reduce risk and improve agility, governance and best practices by standardizing DWA practices across the data and analytics team.
Strategic Planning Assumptions
By 2023, 75% of all databases will be on a cloud platform, reducing the DBMS vendor landscape and increasing complexity for data governance and integration.

By 2024, 50% of new system deployments in the cloud will be based on a cohesive cloud data ecosystem rather than on manually integrated point solutions.

Introduction
Most organizations continue to derive value from data warehouses because they provide support for a broad range of analytics, and thus can meet many analytics requirements. A data warehouse is a central source of data that is well-understood, quality-assured and easy to analyze. Once loaded, the data can be reused in many ways — for known queries or new ones. The logical data warehouse is the modern form that incorporates a data lake and other components, and is often built in the cloud.

This research assumes that data and analytics leaders are not yet familiar with DWA, providing insight to know what DWA is, how it can help, and when it is right to use and when it is not.

Figure 1 illustrates the development of a data warehouse system. Users and developers can collaborate in describing the data warehouse system they want. The DWA tool then generates the necessary system components without manual coding. Note that usually current data warehouse systems adopt the logical data warehouse architecture that incorporates an enterprise data warehouse, data lake and other components (see The Practical Logical Data Warehouse).
Note that the relevance of the strategic planning assumption regarding the move of DBMSs to cloud is that DWA can assist with moving to the cloud by regenerating the system for a new cloud platform. Likewise, DWA can assist with the adoption of cloud data ecosystems by being able to adapt code generation templates.

**Impacts and Recommendations**

**DWA Provides a New Way of Developing Data Warehouse Systems**

To understand the benefits of DWA, the effort required must be compared to normal data warehouse development. Figure 1 shows, in simplified form, that the data warehouse consists of several major components that are built from a large number of development artifacts. These take a great deal of manual effort to produce.
Modern DWA tools can dramatically reduce the amount of work required to build, operate and maintain a data warehouse. DWA helps with change management — and it can enable DevOps, DataOps and other forms of agile DBMS development (see Data and Analytics Essentials: DataOps and Keys to DevOps Success). DWA can also extend beyond central functions of the data warehouse. It can also be used for a wide variety of structural data automation and the building of data pipelines.

Ideally, DWA spans as many stages of a data warehouse life cycle as possible, which may include design, prototyping, testing, production monitoring, maintenance, revisions, etc. Some DWA tools cover most of that; others cover relatively little. Helping across the full life cycle should be a goal of DWA. This automation complements the automation of development and production found in other data management tools, such as those used for data integration, data quality and master data management.

Figure 2 shows a typical DWA software tool architecture. The principal component is the metadata repository. The development team uses metadata to describe the entire system, including all data structures, data flows and presentation data objects. They can use a client tool or a web portal to edit the metadata.
Developers describe the system using metadata, which feeds into code generators. These use the metadata description to generate all the data structures, program logic, and extraction, transformation and loading (ETL) code, thus saving a great deal of manual coding effort. The generators make use of libraries of code templates — templates can be created, customized and reused many times.

Figure 3 shows the typical flow of metadata starting with the user catalog, which defines the users and what they can do. Developers use the DWA tool to align the generated code with the usual site security. The metadata repository has metadata for all of the code and data objects that make up the complete system. When the user or developer invokes a build of the system, the relevant code generator reads in the metadata. The DWA tool then passes the generated code to the relevant execution engine. These can be ETL tools, script engines, executable programs or tasks that can be scheduled.
It should be noted that the DWA tool can also read the metadata into the metadata repository from the outside environment. When developers specify a new database as a source, the DWA tool will read the data definition language (DDL) for that database and import it into its own repository.

For code generation, using customizable templates makes the process more flexible.

As databases adopt more recent and more comprehensive standards of the SQL language, the differences between their SQL interfaces diminish and their expressive power increases. This makes automatic code generation easier.

Some types of metadata are becoming more standardized — for example, cloud service providers are increasingly implementing comprehensive metadata stores.

Recommendations:
DWA Increases Data Warehouse Developer Productivity

Working with a DWA platform requires developers to work at a higher, more abstract level of definition than the manual coding they are used to. This can feel strange, but most will rapidly adjust. Data and analytics leaders should plan for a transition period and training to ensure that developers can adopt this new development mindset. Figure 4 illustrates developers working at this higher level.

Figure 4: Working at a Higher Level of Abstraction

- Become familiar with DWA tools and methods to be in a good position to assess them.
- Identify whether all or part of your data warehouse development can benefit from DWA.
Referring to the left side of Figure 4, DWA developers are using a DWA tool and defining the system and linkages between the components in metadata. Once defined, the DWA tool will then automatically generate all the components shown in the middle of the diagram. On the right is the alternative manual approach to generating those same components. Doing this, then integrating and testing them (and doing this manually and repeatedly), requires more effort.

DWA tools can also apply automation into development of data lakes and their object stores. Assisting with the deployment of data warehouses in the cloud is another area that these tools are maturing into.

Few companies can implement all their potential requirements — limits on resources typically preclude this. Therefore, a DWA tool is often used as a productivity enhancer, which helps to meet more requirements and thus deliver more benefits. The dependence on external staff, such as contractors, can also be reduced using DWA. (See Note 1 for a nonexhaustive list of vendors in this market.)

Recommendations:

- Compare working practices enabled by DWA with the equivalent manual effort; ensure that this includes a full life cycle with multiple rounds of system enhancements — not just initial development.

- Examine requirements and any existing backlog to assess what level of ongoing benefit you would expect to gain.

DWA Enables Agility

DWA enables the agile development of analytical systems and pipelines in several ways, as shown in Figure 5. Many of these are within the familiar DevOps process.

Using DWA, developers can do the essential upfront work more quickly. It also reduces the temptation to take dangerous shortcuts that will cause technical debt and other problems later.
Developers gain agility through the ability to generate code and — equally important — the ability to regenerate it to make changes. Being able to make incremental changes to the data warehouse being developed or the development process, and to move from one target system to another quickly, are key strengths of DWA tools.

Aside from generating code for the original state and the new state, the DWA tool can also generate special code to transition from one state to the other; however, be aware that not all DWA solutions have this advanced functionality. For example, consider the process of adding columns to a presentation layer data mart. The DWA tool generated the data description for the original mart, then the ETL loaded it. Done manually, the developers will also need to write some special, one-off code to unload the mart, reshape it and then reload the data in its new form, merging data for the new columns. Some DWA tools can automatically generate this transitional code too. This is faster and much more productive than having to do so manually.
Over the lifetime of a data warehouse, developers will repeatedly perform this type of process. The principle is to build once and then repeatedly reuse. The same applies to the generation of logic to enable testing — test code harnesses can also be generated. This also applies to database migration and system performance improvements made through the adoption of new DBMS features where the system can be regenerated to start using the new features.

Various different data modeling methods can be used. Most DWA tools assist with data vault data modeling, which needs a well disciplined approach as it uses a greater number of physical tables. DWA tools can act as a framework to supply the necessary discipline. Often, they also allow the choice of third normal form modeling or dimensional modeling. Some DWA tools also allow you to switch between different modeling methods by regenerating the system code according to the newly desired method.

Regenerating the code for a new target platform (or feature) is much faster than manually migrating it. The development team must still test the new system, but this too can be semiautomated using DWA.

**Recommendations:**

- Determine where the greater agility enabled by a DWA tool can be used to improve your development process through enhanced DevOps principles — not just as a productivity tool.
- Take into consideration that DWA is also a valuable tool for risk reduction, governance and adoption of best practices.

**Evidence**

Interactions with Gartner clients and vendors throughout 2021 informed this research.

**Note 1: Vendors in the DWA Market**

Vendors in this market include, but are not limited to:

- AnalyticsCreator Solutions
- BI Builders (Xpert BI)
- Concentra Analytics (TrueCue)
- Datavault Builder
These organizations are provided for illustrative purposes only. This is not an exhaustive list of examples in the DWA market nor a Gartner endorsement of the vendors or their offerings.

**Recommended by the Author**

Some documents may not be available as part of your current Gartner subscription.

*Automating Data Warehouse Development*

*Assessing the Capabilities of Data Warehouse Automation (DWA)*

*Data and Analytics Essentials: DataOps*

*The Practical Logical Data Warehouse*

*6 Things to Get Right for the Logical Data Warehouse*

*Quick Answer: What Is Data Warehouse Automation?*