The Sedona Conference Working Group Series

The Sedona Conference

Database Principles Addressing the Preservation and Production of Databases and Database Information in Civil Litigation

A Project of The Sedona Conference Working Group on Document Retention and Production (WG1)

July 2025
Public Comment Version
Submit comments by Aug. 25, 2025,
to comments@sedonaconference.org

Database Principles Addressing the Preservation and Production of Databases and Database Information in Civil Litigation

A Project of The Sedona Conference Working Group on Electronic Document Retention and Production (WG1)

JULY 2025 PUBLIC COMMENT VERSION

Drafting Team Leader

Scott Clary

Drafting Team Members

Jeffrey Bannon   Courtney Fletcher

Lilith Bat-Leah   Margaret Malloy

Gregg Parker   Anthony Pastore

Chuck Rothman   Jonathan Swerdloff

Steering Committee Liaison

Meghan Podolny

Staff Editor: Craig Morgan

The opinions expressed in this publication, unless otherwise attributed, represent consensus views of the members of The Sedona Conference Working Group 1. They do not necessarily represent the views of any of the individual participants or their employers, clients, or any other organizations to which any of the participants belong, nor do they necessarily represent official positions of The Sedona Conference.

We thank all of our Working Group Series Annual Sponsors, whose support is essential to our ability to develop Working Group Series publications. For a listing of our sponsors, just click on the "Sponsors" navigation bar on the homepage of our website.

REPRINT REQUESTS:

Requests for reprints or reprint information should be directed to
The Sedona Conference at info@sedonaconference.org.

Visit www.thesedonaconference.org

Preface

Welcome to the July 2025 Public Comment Version of The Sedona Conference's *Commentary on Addressing Databases in Civil Discovery* a project of The Sedona Conference Working Group 1 on Electronic Document Retention and Production (WG1). This is one of a series of Working Group commentaries published by The Sedona Conference, a 501(c)(3) research and educational institute dedicated to the advanced study of law and policy in the areas of antitrust law, complex litigation, intellectual property rights, and data security and privacy law. The mission of The Sedona Conference is to move the law forward in a reasoned and just way.

The mission of WG1, formed in 2002, is "to develop principles, guidance and best practice recommendations for information governance and electronic discovery in the context of litigation, dispute resolution and investigations." The Working Group consists of members representing all stakeholders in the areas of eDiscovery and electronic records management.

The *Commentary* is the second edition of the September 2014 publication entitled *Database Principles*. The editors and the WG1 Steering Committee decided that a title change was necessary to avoid confusion with WG1's flagship publication, *The Sedona Principles*, addressing eDiscovery in general, currently in its third edition and cited in more than 275 state and federal court decisions.

This *Commentary* has a lengthy history, reflecting the rapid evolution and complexity of the underlying information technology. The WG1 Steering Committee launched a "brainstorming team" in July 2021 to explore updates to the 2014 publication. Proposals were the subject of dialogue at the WG1 Annual Meeting in October 2021 and again at the Midyear Meeting in April 2022. A draft was presented at the Annual Meeting in October 2022 and was the subject of further dialogue at the Midyear Meeting in April 2023, resulting in an updated draft in 2024.

This *Commentary* represents the collective efforts of many individual contributors. On behalf of The Sedona Conference, I thank the drafting team: Scott Clary, Jeffrey Bannon, Courtney Fletcher, Lilith Bat-Leah, Maggie Malloy, Gregg Parker, Anthony Pastore, Chuck Rothman, and Jonathan Swerdloff. The drafting process for this *Commentary* has also been supported by the Working Group 1 Steering Committee.

The statements in this *Commentary* are solely those of the individual members of the Working Group. They represent a consensus of recommendations that the contributors and reviewers feel will be helpful to organizations, their counsel, legal technologists, and courts to avoid or reduce discovery disputes, and are not necessarily the views of any particular individual nor the endorsement of any particular approach.

Please note that public comment on this edition of the *Commentary* is open through Aug. 25 and suggestions for improvements are welcome. After the deadline for public comment has passed, the drafting team will review the public comments and determine what edits are appropriate for the final version. Please send comments to comments@sedonaconference.org.

We encourage your active engagement in the dialogue. Membership in The Sedona Conference Working Group Series is open to all. The Series includes WG1 and several other Working Groups in the areas of artificial intelligence, cross-border discovery and data protection laws, international data transfers, data security and privacy liability, trade secret protection, and patent litigation management. The Sedona Conference hopes and anticipates that the output of its Working Groups will evolve into authoritative statements of law, both as it is and as it should be.

Kenneth J. Withers
Executive Director
The Sedona Conference
July 2025

Table of Contents

Executive Overview

The Sedona Conference Working Group on Electronic Document Retention and Production has developed principles in this *Commentary on Addressing Databases in Civil Discovery*. Within this *Commentary*, we offer several practical suggestions that we believe clarify the obligations of both requesting and producing parties, and simplify discovery in matters involving databases and information derived from databases. We recognize that the specific facts of a litigation matter, combined with the implementation of relevant databases likely will raise additional retention and production issues not explicitly covered by this paper. Even so, we believe that the groundwork laid by this *Commentary* will provide valuable guidance to litigants facing novel issues of database retention and production.

It is important to set reasonable expectations for the production of database information, and thus, an overarching theme of this *Commentary* is that communication—between database management professionals and the attorneys who are asking them to search and export litigation-specific information, as well as between requesting and producing attorneys—is critical when working with databases. Many common disputes about issues such as the production format of data can be reduced or even eliminated through better dialogue between litigants.[1] We also find that better communication naturally will reduce blunderbuss requests for databases that typically encompass irrelevant or inappropriate information, or the production of terabytes of useless, undifferentiated data.

Our *Commentary* is divided into four discrete sections. Following a brief Introduction in Section I to databases and database theory, Section II addresses how The Sedona Principles, which pertain to all forms of ESI, may be applied to discovery of databases. Section III proposes six Principles that pertain specifically to databases and provides commentary to support our recommendations. Section IV is an appendix covering the most common database platforms used in business today.

As database technology continues to evolve, we acknowledge that *Commentary on Addressing Databases in Civil Discovery* will need to be revisited regularly to ensure that its guidance remains topical. At the same time, we believe that this *Commentary* lays a foundation that will be valid both today and in the future for developing effective and practical solutions in this sophisticated area of the law.

[1] The Sedona Conference, *The Sedona Conference Cooperation Proclamation*, 10 SEDONA CONF. J. 331 (2009 Supp.).

The Sedona Conference Database Principles

The Sedona Conference Working Group on Electronic Document Retention and Production (WG1) has been studying issues about the discovery of database information in civil litigation and has developed the following *Principles* addressing the preservation and production of databases, *The Sedona Conference Database Principles*.

1. **Scope of Discovery**
   Absent a specific showing of need, a requesting party is entitled only to database fields that contain relevant information, and give context to such information, and not to the entire database in which the information resides or the underlying database application or database engine**.**

2. **Accessibility and Proportionality**
   Due to differences in the way that information is stored or programmed into a database, not all information in a database may be equally accessible, and parties should therefore apply proportionality to each component of a database to determine the marginal value of the information to the litigation and the marginal cost of collecting and producing it.

3. **Use of Test Queries and Pilots**
   Parties should use objective information, such as that generated from test queries, pilot projects, and interviews with persons with relevant knowledge to ascertain the burden and benefits to collect and produce information stored in databases, and to reach consensus on the scope of discovery.

4. **Validation**
   A responding party should use reasonable measures to validate that its collection from the database is both reasonably complete and did not inadvertently modify the ESI.

5. **Data Authenticity and Admissibility**
   The proper validation of collection from a database does not automatically make the substantive information stored in the database authentic, admissible or true. These are separate issues that need to be analyzed by the appropriate decision-makers.

6. **Form of Production**
   The way in which a requesting party intends to use database information is an important factor in determining an appropriate format of production.

I. INTRODUCTION

Disputes over the discovery of information stored in databases are increasingly common in civil litigation. Part of the reason is that more and more enterprise-level information is being stored in shared, searchable data repositories (structured data)[2] rather than in discrete electronic files (unstructured data). Another factor is that the diverse and complicated ways in which database information can be stored has made it difficult to develop universal best-practice approaches to requesting and producing information stored in databases. These storage factors include data residing in cloud environments, or in third-party software as a service environment, to name a few. The procedures that work well for simple systems may not make sense when applied to larger systems that manage big data sets with real-time streams. Similarly, data retention policies vary widely for different types of databases, from very short lifespans of data that can be measured in minutes or seconds, to indefinite retention. (It is not uncommon for databases to have no purge or delete routines.).

A. How Is ESI Stored in Databases?

Successfully working in a discovery context with databases and the structured data found in them requires a basic understanding of this form of electronically stored information (ESI) as it functions in the ordinary course of business.

This commentary is intended to address considerations that may apply when databases[3] contain discoverable structured data, rather than unstructured data. Structured data tends to have the following characteristics:

- Logical entities[4] are decomposed into their constituent data elements (known as fields or records) at a highly granular level;

[2] The Sedona Conference, *The Sedona Conference Glossary: E-discovery & Digital Information Management*, 21 SEDONA CONF. J. 375 (5th ed. 2020), ("The Sedona Conference Glossary"), defines structured data as: "[d]ata stored in a structured format, such as databases or data sets according to specific form and content rules as defined by each field of the database." *Id.* at 356.

[3] *Id.* at 291, defines a database as: "A set of data elements consisting of at least one file or of a group of integrated files, usually stored in one location and made available to several users. …Computer databases typically contain aggregations of data records or files. …"

[4] *Id.* at 333, defines logical entities as: "An abstraction of a real-world object or concept that is both independent and unique. Conceptually, a logical entity is a noun, and its relationships to other entities are verbs. In a relational database, a logical entity is represented as a table. Attributes of the entity are in columns of the table and instances of the entity are in rows of the table. Examples of logical entities are employees of a company, products in a store's catalog, and patients' medical histories."

- Individual data elements are stored in specific assigned logical and physical areas within a series of files (or a single fielded table or a text delimited file[5]);
- These data elements are linked to each other by internal mechanisms, interpretable by the database software;
- These links or relationships may involve metadata elements stored within the database, in addition to the data elements of the logical entity; and
- Once properly assembled and formatted (e.g., in the form of a report or table), structured data is often readily understandable.

For example, in the case of a simple invoice being stored in a relational database, the logical entity invoice might consist of customer name, customer address, item ordered, cost of item, etc. These data elements themselves consist of more granular data elements. For example, customer name could be further decomposed into customer first name, customer middle initial, and customer last name. Similarly, item ordered could be decomposed into item description, SKU number, and price. These data elements are commonly placed in structures called tables, which are used to organize the information, as defined further below.

By contrast, unstructured data[6] tends to have the following characteristics:

- Stand-alone ESI consisting of a self-contained file or document (examples include MS Word, MS Excel, Adobe PDF, etc.);
- Generally does not require any highly technical knowledge to understand or use an individual file or document containing unstructured data; and
- Both the creation or selection of information to be included in the file or document and the way that information is formatted for display are left to the discretion of the creator of the file or document containing unstructured data.

[5] *Id.* at 379, defines text delimited file as: "A common format for structured data exchange whereby a text file contains fielded data, separated by a specific ASCII character and also usually containing a header line that defines the fields contained in the file."

[6] *Id.* at 385, defines unstructured data as: "[F]ree-form data that either does not have a data structure or has a data structure not easily readable by a computer without the use of a specific program designed to interpret the data; created without limitations on formatting or content by the program with which it is being created. Examples include word processing documents or slide presentations."

Structured data may be found in contexts that you might otherwise expect to contain unstructured data, such as email database systems[7] or websites (e.g., Domino/Lotus Notes, or WordPress). Conversely, unstructured data may be embedded in structured data (e.g., a customer invoice might be stored in a database column as a .pdf file). This Commentary is intended to address situations where structured data is discoverable, including when structured data is contained within a stand-alone file or a document is needed for context.

For structured data in a database, individual data elements or fields—each of which needs be accessed separately for relevance—must be assembled and viewed in context to be understood. Databases, however, impose strict rules that define how information can be entered, stored, and retrieved. For example, a particular database might store a customer's name, John Q. Smith, as three discrete elements—first name (John), middle initial (Q), and last name (Smith)—each in separate data fields. Unlike the unstructured file, these separate elements must reference each other to be recalled and displayed as a whole name. Each database may have its own unique rules for storing and recalling elements of information. Additionally, different applications (even those written on the same type of database system) may be designed differently and may store a whole name (for example John Q. Smith) in a single field without dividing it further.

End-users commonly think of database information in terms of records they query, retrieve, and view. Although a database record may be the closest intellectual analog to a document within a database, records consist of separate data elements that may be stored in a number of ways within a database, such as in multiple tables, or across multiple databases. Thus, a record may not exist until actions by a user instruct the database application to assemble specified fields for display. Accordingly, a database record is not always an appropriate granular level of information to respond to a discovery request. At various times, key information may be found in a single data field, in a record made up of a set of selected fields, in a table containing a pool of records, or in a report that extracts discrete fields of information from multiple tables. Thus, the extraction of responsive information from databases may often require specialized business or technical knowledge. Rather than requesting or producing a file or document, each party should carefully consider what it means to produce responsive information. For example, certain spreadsheets may not display information as it was maintained in the ordinary course of business if not produced with all linked spreadsheets.[8]

For instance, using the simple example of the organized collection of customer invoices, the customer record might be defined as a set of fields, composed of the following fields:

[7] Although the email message content itself is unstructured, emails are accompanied by metadata in assigned fields, including, but not limited to, the sender, recipient, date, and time. The message content and metadata elements are stored together in an email database system, comprising an email record. The email database system stores individual email records, imposing the same storage format across all individual email records.

[8] Although linked spreadsheets are not technically a database, this *Commentary* may provide general guidance for discovery of structured data.

FIRST NAME:
MIDDLE INITIAL:
LAST NAME:
STREET ADDRESS:
SUITE NUMBER:
CITY:
STATE:
ZIP CODE:
TELEPHONE NUMBER:
FAX NUMBER:
EMAIL ADDRESS:
COMMENTS:

Hundreds or thousands of such customer records may be stored in the database, with the elements for each customer arranged in a data table or a set of data tables and sub-tables, depending on the complexity of the database. A record from this database, showing the information for a single customer, may appear to the user issuing a query to the database as a collection of selected fields in a pre-determined format for that query, perhaps as a mailing label with only the name and address, or perhaps as a complete dossier with the contact information and a record of past transactions for that one customer derived from related databases. In addition to requesting a record from this database through a query, a user may ask for a report based on selected fields across many records, for instance the names of all marketing contacts within a particular state, ordered numerically by zip code and then alphabetically by last name.[9]

Databases systems tend to be highly unique and customized to support a specific task or system owner. Thus, in addition to the context typically required to understand the significance of a traditional document, the ability to fully understand the unstructured data within a database requires knowledge of data relationships, what the information represents, and how it was generated. Without this information, analyzing databases is akin to seeing a thousand-piece jigsaw puzzle without an illustration that shows the final completed puzzle. The jigsaw puzzle can be assembled, but only with great effort and with low efficiency.

To add to the jigsaw puzzle, often the database system is now hosted in the cloud by a third -party entity. These cloud-hosted environments can mimic the classic database servers that many e-discovery professionals and database administrators are familiar with navigating. The cloud-based database environments can also be a proprietary environment with custom software as a service application

[9] This description of a database with its structured data should be distinguished from the term "data compilation," introduced in the 1970 amendment to Rule 34 the Federal Rules of Civil Procedure, long before the advent of the desktop PC and off-the-shelf database software. That term was intended to encompass all of what we think of today as "electronically stored information," and was occasionally used by courts interchangeably with the term "database," even though the "records" in such "databases" may have included unstructured data. *See, e.g.*, *Fauteck v. Montgomery Ward & Co.*, 91 F.R.D. 393 (N.D. Ill. 1980) (machine-readable employment records).

built around the environment where vast big data volumes are live streamed into the database from sources around the world every second. The data streams can be aggregated from IoT devices (e.g., smart watches, refrigerators, security cameras), automobiles, social media applications, and televisions, to name a few.

B. Components of a Typical Database System

Database systems typically consist of the following elements:

- Database application: A software program or programs, usually designed for a specific purpose, and usually providing a "higher-level" view of the data (often through a graphical user interface) that conceals the complexity of data decomposition and data location. Multiple applications may point to the same database.

- Database management system (DBMS): The software program that stores and retrieves data at a basic level and interfaces between the applications and the database files. For example, a database management system may enforce rules pertaining to data such as only allowing storage of numbers in a telephone number field or ensuring that all invoices pertain to a customer.

- Set of structured tables or files: These contain the substantive data, often in a vendor-specific format.

Confusion can arise when parties use the same terminology to describe all three components of a database system.

The individual parts of a database system may themselves be composed of multiple parts. A database management system may be composed of multiple software programs, including storage engines, that collectively provide core database functionality in a given hardware and operating system environment. The database application may be composed of tens—or hundreds—of individual programs. The database storage file that typically contains the information relevant to a specific legal dispute may be a single file, but more commonly, it is composed of multiple separate data storage files in multiple locations. Large storage systems may be composed of hundreds of separate data files and may reside on many computers.

C. Assessing Relevance for Databases and Database Records

For the reasons provided in sections A and B above, the legal team often will require the assistance of individuals with technical and business expertise to assess what information within a database system is responsive to a particular matter. Although a database system may contain relevant, even critical, information, it also may contain information that is irrelevant or only tangentially related to the issues in a particular case. For example, the financial accounting system used by a large company may contain thousands of different data tables and tens of thousands of data fields. In most cases, however, only the substantive information contained in a small number of tables or fields will contain information of direct relevance to a legal dispute, unless the dispute relates specifically to the design or performance of the system. Thus, working successfully with a database system requires understanding how information is organized within a database and the relevance of the various fields to the issues. Database design may include detailed documentation reflecting the database design, entity and relationship schemas, and basic description of the data in the components of the database. This information is helpful when working with the parties to identify the portions of the information in the database that are potentially responsive.

To identify the data that might be relevant in a particular matter, the legal team must understand the core issues of the case, the facts that might prove or disprove liability, and the factors that might be useful in establishing or refuting damages. Different types of cases will require different types of information and will make use of database information in different ways.

D. Preservation of Databases

A party is obligated to take reasonable steps to prevent the deletion or modification of information in its possession, custody or control that it knows or reasonably should know is relevant to pending or reasonably anticipated litigation. This obligation applies to databases but differs from preservation of unstructured ESI in a number of important ways. Preservation of information contained in databases usually requires expertise of database system or application administrators. For certain information in databases that is not overwritten (and is essentially aggregated), it is reasonable to preserve the data in place, but for other dynamic data that is not stable it may not be technically possible to preserve the data in place. For instance, if the data is volatile (subject to being programmatically changed or deleted) or if the database system or application has enforced retention periods that for technical reasons cannot be readily suspended or interrupted, then it may be advisable to copy the specific responsive information to a separate secure location in a manner that protects that responsive data. Because of the expense of production, restoring, and interpreting backups from tape or disk, preservation by means of backups should only be used in situations where there is no other reasonable means of preservation. One thing that is consistent across databases and unstructured data is that responding parties are only obligated to take steps to preserve the information that is relevant to the matter and not all data within the database or in the data source.

E. Collecting and Producing Database Information

Differences in ways that database information and individual documents are organized also require different approaches and tools in the traditional discovery tasks of collection, review, and production. Unlike loose documents, database information does not fit neatly into standard document collection protocols. It is in the interests of both requesting and responding parties to avoid over-production of information. Other than situations where a large portion of a given database is responsive, it may be best practice to collect that responsive data by saving a copy of a subset of the database information to a separate location, such as a specifically designed table, a separate database, or a text delimited file by means of a query or report. In some cases, a pre-existing (canned) query or report may exist that can be used for this purpose. In other cases, a custom-created query or report will need to be used.[10]

Assuming that one can create a separate copy of a subset of relevant information from the database, the format by which this will be produced should be considered. Unlike text delimited files, a given database format will often not be readable by other software. Therefore, both parties should communicate early about the format for production so that the ESI is reasonably usable by the receiving party in accordance with Rule 34.

These uniqueness and customization issues preclude the use of generic ESI collection tools to capture relevant information within a database. Consequently, the process for understanding and retrieving the data from databases can require significant hands-on involvement by the database managers as well as database users to educate the legal team about the contents and structure of the database in question. This process is often matter-specific and potentially labor-intensive. Conversely, the very nature of the database is to put information into an organized data set that can be harnessed for efficient data retrieval. Often, transparency and cooperation[11] between the parties as it relates to the type of information and the structure of the database can help accomplish a proportional but responsive collection of information from a structured data set.

Certain specific types of contextual information are commonly requested and produced from databases. These include:

- Field names which may or may not help the requesting party understand the contents of each field. Note that field names and field contents may not necessarily be related, as in databases that have been in use for some time or whose primary design objectives have changed.

- Field values and codes which define any abbreviations stored in data fields. Field codes, whether abbreviated or not, may require further context to convey their meaning to a requesting party. For example, the code SG that is stored in the Product Category field might require both translation to Sporting Goods and a further description of what this term encompasses within the organization. Field value translations and/or associated lookup tables may be critical to understanding accurately the content of the data file, and a responding party should provide this additional information if necessary.

- Input constraints that describe the allowable and/or expected values in a field. Common examples of field input constraints include numeric-only limits, state code abbreviations, and ZIP code validation. Understanding field constraints can explain why the data has been standardized in a specific way. Conversely, knowledge of these input constraints can make it

[10] Note that if a text delimited file is produced and the format does not have column headings, then it is also generally necessary to produce metadata to explain the fields in the text file.

[11] The Sedona Conference*, The Sedona Conference Cooperation Proclamation*, 10 SEDONA CONF. J. 331 (2008).

easy to check data production for errors; abnormal field values in the production may indicate that there were errors in process used to extract and prepare the data for production.

- Auto-filled fields such as username or time stamps are populated automatically by the system and without human intervention. These fields may be valuable validation tools in the ordinary course of business, as they are unlikely to contain human data entry errors, and they may have similar value in authenticating database information for possible evidentiary use. A requesting party may find it valuable to request the identification of these fields, along with the rules or programming logic used to populate them.

Information contained in databases may be the best source for establishing certain facts in a legal dispute. Information stored in this format also may be useful, if not essential, for analyses such as sorting, calculating, and linking to answer quantitative questions presented in a case. In contrast, documents such as individual email messages and free-form electronic word processing and presentation documents are not easily calculated or sorted based on their content, though they may better answer certain qualitative (as opposed to quantitative) questions than database information. Information extracted from databases is often used by accounting or economics experts on behalf of litigants, who use the quantitative conclusions of these analyses to support their legal positions.

In some situations, the producing party may not have access to the DBMS or the schema and may only be able to access the database via the user interface of an application. This does not necessarily mean that the data is not reasonably accessible, and parties should discuss a reasonable and proportional production methodology under the circumstances.

F. Potential Use of Database Information by a Requesting Party

An important consideration in how database information should be requested and produced in civil litigation or regulatory discovery is the manner in which the requesting party intends to use the information. Without such mutual understanding, databases and database information may be produced in ways—even electronic, machine-readable formats—that are not suitable for the requesting party's needs. A requesting party may use structured ESI in a variety of ways, including, but not limited to: (1) reviewing specific historical transactions and records; (2) developing an archive of information that can be queried as might have been done in the ordinary course of business; or (3) developing new analyses of the information that are based on a current, not historical, understanding of the data. The anticipated use of the data will drive the discussion regarding the most appropriate production format for structured ESI from a database system.

Reviewing historical information typically requires the simplest production format of these three potential uses. If the parties are interested in discrete transactions or events, a simple query or review of the data to isolate relevant records may be sufficient. A simple example of this use would be querying a database for information regarding a specific invoice. Depending on the volume of information required for this use, database information can be produced in a number of different production formats, possibly even those that do not preserve the fielded nature of the information. Simple canned reports displaying the requested information may be adequate, and such reports sometimes can be exported into standard electronic formatted files, such as Microsoft Excel, or Comma Separated Value (.CSV).

However, developing an archive of relevant information that can be queried as might have been done in the ordinary course of business may require a more elaborate production format. For example, if the dispute involves all invoices and other interactions with a particular customer, relevant information may include a large volume of invoices and other accounting information, as well as standard reports that were generated or used by key players in the dispute as the basis for decisions involving that customer. Sometimes, the requesting party also may want to replicate standard reports that were used by the producing party, but with altered parameters, such as generating reports based on quarterly instead of annual data.

For purposes of deciding a production format, one key consideration is whether the requesting party will need to generate various alternative reports using a variety of search parameters. If so, then it is likely that the requesting party will need to receive not only the source data, but also a means to edit the canned reports, or create new reports. However, when the relevant information is contained in only a few set reports, the producing party may be in the best position to generate and produce the specific reports for the requesting party.

The need of a requesting party to develop new queries and reports to analyze the data from an existing, and particularly legacy, database system can raise the greatest challenges to identifying and implementing a useful production format for database information. For example, when a requesting party has a legitimate need to develop an independent analysis or show the significance of viewing the data in a certain way, responsive data must be provided in a format that supports the legitimate intended use. As such, the requesting party must make reasonable efforts to work with the responding party to ensure that structured ESI extracted from a database is produced in an appropriate, reasonably usable format. This can be a complicated process for the producing party, particularly if the requesting party seeks the underlying data in a format in which it ordinarily has not been stored. When such situations arise, the parties should consider the scope of the request and the cost and effort required to collect and produce the information from the database in a reasonably usable format.[12]

The data analysis undertaken by a requesting party can range from simple data accumulations, such as total sales in a given time period, to complex time trending that reveals specific patterns in the data. Often, the requesting party will need to create custom reports or new tables to support these analyses. To ensure the accuracy of the underlying source data on which these analyses are based, at times it may be necessary to produce operational manuals, schematics, or other ancillary documentation that is required for the requesting party to correctly assemble the data.

[12] The Sedona Conference, *The Sedona Conference Database Principles Addressing the Preservation and Production of Databases and Database Information in Civil Litigation*, Database Principle 2, Accessibility and Proportionality, 15 SEDONA CONF. J. 205 (2014).

Creating new analyses of information contained in a database often expands a discovery request beyond the immediate fields that contain the substantive information at issue. For example, a call center application may have components that help manage the workflow between the agents. This may include external logs that track who participated in a particular call, how the call was processed, and its ultimate disposition. Even if the responding party does not routinely look at all this stored information, if there is a question as to how the responding party managed its calls, then a requesting party may reasonably want to analyze this data, including the internal system fields that are not visible to the user that tie these disparate data elements together. Therefore, it is critical for the parties to confer as to the scope and format of the information to be produced.

A final consideration with respect to the requesting party's need to perform new analyses on the structured ESI is the extent to which the requested information can be introduced as substantive evidence in court. While the traditional approach for introducing this type of electronic evidence is through a testifying expert, some testifying experts may not be qualified to manipulate the underlying data to create the analysis that may form a partial basis for their conclusions. Certain experts may instead work with one or more technicians who serve as the interface between the data and the testifying expert. At this time, there are no standard practices with respect to these data technicians, and it is unclear to what extent their activities must be validated or whether they themselves must be available to testify as fact or expert witnesses to meet the evidentiary requirements. Further, such data processing has at times introduced questions regarding the accuracy and admissibility of analyses, even though they are based on the original data produced in discovery by an opponent.

G. Locating Specific Database Information through Queries

Counsel should adequately communicate with the information technologists, database users, or other client representatives responsible for the database systems to determine the most efficient way to locate the responsive data. Those who are responsible for identifying relevant database information may need to rely on search tools, particularly for ESI within a larger database or database system. Three basic types of tools are available for this task: (1) built-in search functions relying upon an internal database index; (2) search functions that search database content in real-time (non-indexed) searches; and (3) third-party tools that develop their own indices or search existing data tables using alternate search algorithms. However, it should be noted that the Information Technology (IT) departments in many large organizations require that such third-party tools be comprehensively tested before installation or use to ensure that data integrity and operational functionality are not impaired. In such situations, the testing protocols can be quite rigorous and time consuming, thus potentially affecting the practicality of this third option.

Database indices[13] can be used to speed up queries against database data. Because database indices typically reference only a subset of the data fields that exist within a database, parties may need to assess the value of using additional technology to conduct broader searches that access more or

[13] *The Sedona Conference Glossary* defines index as: "Database fields used to categorize and organize records. Often user-defined, these fields can be used for searching for and retrieving records."

additional information within a database. However, such "database-crawling" tools can significantly impact the speed at which a database processes transactions. In considering whether such supplemental measures are required, the parties should weigh the likelihood that the search will provide useful additional information against the burden that this approach would place on the responding party, both in terms of litigation costs and potential business disruption. This analysis can be very fact-specific and requires that the parties engage in an open and well-informed dialogue.[14]

H. Databases and Database Information in a Third Party's Custody or Control

It is increasingly common for companies to outsource some or even all their IT functions to third parties—including the storage and management of all ESI, which includes database information. For example, many companies outsource their payroll function to another company that maintains some, if not all, of the detailed information regarding payroll on their databases and systems. In certain situations, information managed and maintained by these third parties could become relevant in a legal dispute and fall under a legal hold. In addition, while the substantive data sought by a requesting party may be deemed within the responding party's possession, custody, and control, there may be ancillary data or metadata necessary for full understanding of the substantive data. Such information, like field structures or metadata, may be in the hands of a vendor or service provider, requiring a subpoena under Rule 45 to obtain. While the situation of potentially relevant data being stored at a third-party location outside the possession, custody, or control of a litigant is not new or even limited to ESI, discovery of database information stored in a third-party repository can involve a complex mix of competing rights and obligations that may require court intervention to resolve.

- When data is housed by third parties (e.g., cloud computing), it can complicate the legal and technical issues related to data preservation and production. These issues are beyond the scope of this Commentary, but some of the important issues to keep in mind are:

- Whether a party can legally obtain requested database information from the third party and the costs involved, which may be governed by the terms of a service contract

- The extent to which the requested data may be co-located with data of other non-parties, and the difficulty of extracting only the requested data

- The extent to which proprietary information, software, or equipment of the third party is required to understand or use the requested data

[14] *See, e.g.*, *Soto v. Genentech, Inc.*, No. 08-60331-CIV, 2008 WL 4621832 (S.D. Fla. Oct. 17, 2008) (producing party failed to provide sufficiently detailed information to support its burdensomeness argument as to the time and effort required to compile certain relevant information stored in databases). *See also FDIC v. Brudnicki*, No. 5:12-cv-00398-RS-GRJ, 2013 WL 2948098291 F.R.D. 669 (N.D. Fla., Panama City Division, June 14, 2013) (rejecting the argument that proposed database search protocol requiring parties to collaborate in creating search terms was unduly burdensome and permitting modest cost-shifting consistent with traditional paper cost-shifting).

- The extent to which the integrity or management of the data by the third party is itself a relevant issue in the litigation

- Whether in any litigation, it is more appropriate or efficient to request an opposing party to produce the data under Rule 34, or request a third party to produce the data under Rule 45

II. APPLICATION OF THE EXISTING SEDONA PRINCIPLES TO DATABASES AND DATABASE INFORMATION

Since 2003, The Sedona Principles: Best Practices Recommendations & Principles for Addressing Electronic Document Production[15] has provided guidance to the legal community for the preservation and production of all forms of ESI, including databases. In Section III of this Commentary, we propose six new Database Principles that specifically address the issues associated with databases and database information. However, discussion of how the existing Sedona Principles (Third 2018), particularly principles 3, 5, 6 and 12, apply to the discovery of databases and database information is instructive.

A. Sedona Principle 3: The Early "Meet and Confer"

As soon as practicable, parties should confer and seek to reach agreement regarding the preservation and production of electronically stored information. Sedona Principle 3 is especially applicable in the context of database discovery because of the complicated technical and logistical questions raised by the storage of information in database systems. Database discovery may entail some of the most expensive and complex discovery in a litigation matter, and meaningful conversations between the parties early in the litigation can substantially reduce confusion and waste of resources. It may be in the best interest of the parties to meet and confer regarding the specific fields that contain relevant information, and the specific exports and production format.

By addressing issues related to the preservation and production of information stored in databases as early as possible, parties can resolve easier questions and make progress on resolving more difficult ones. Sharing technical information also may benefit a responding party by educating the requesting party as to what information exists. Such early disclosure can help a responding party avoid wasting resources looking for data that does not exist or that the requesting party does not actually intend to use. Similarly, early discussion may identify specific costs or burden points that can be resolved relatively easily. For example, an ongoing preservation[16] would involve continually preserving every change to dynamic data fields, can be time consuming, prohibitively expensive, and may not be practical in certain database systems. Advised of this, a requesting party may find that it needs

[15] The Sedona Conference, *The Sedona Principles, Third Edition: Best Practices, Recommendations & Principles for Addressing Electronic Document Production*, 19 SEDONA CONF. J. 1 (2018). (hereinafter *The Sedona Principles*).

[16] Ongoing preservation is not only of historical information that pertains to certain conditions, but also of any new information coming into the system pertaining to those same conditions.

only a single snapshot of that information, sparing the responding party unnecessary preservation costs.

1. Redactions, Omitted Data Fields, and the Inadvertent Production of Privileged and Other Protected Data

While a database that logs the use of electronic key cards for entrance into a building is unlikely to contain any attorney-client communications or work-product materials, some databases may contain granular information that requires special protection. For example, a database may contain personally identifying information, such as Social Security numbers, of the people using the key codes. Similarly, a database system that is used to manage a workflow for creating and publishing promotional material may store comments from the in-house or retained legal counsel regarding the materials that fall under the attorney-client privilege. Such privileged notations may be placed in discrete "attorney notes" fields that could be isolated, or they could be mixed with non-privileged comments in free-text data fields.[17]

Early conversations between counsel regarding the existence of protected database information and how that database information should be treated can reduce costs and burden on both sides. For example, both sides may agree that the responding party need not disclose its employees' or third parties' Social Security numbers, thus sparing the requesting party the need to set up complicated protective structures to comply with privacy laws or regulations. However, that may not always be possible. Using the earlier example of privileged communications that may be mixed with other free-form notes, the requesting party still may seek production of this field, with any privileged communications redacted and logged. Under such circumstances, the responding party may be required to budget for and execute a review of the database content, creation of a database-specific privilege log, and development of a protocol that clearly identifies the redaction of this content without otherwise disturbing the integrity of the rest of the data being produced.

It is good practice to discuss the topic of redaction early in discovery in general, and even more so with redaction of database information. Redaction of database information can take two basic forms: (1) not producing a field of information; and (2) overwriting some or all information in a data field so that the requesting party can see that information had been stored in the field. Early discussion can yield agreement on the type of redaction applied to protected information, such as replacing text with strings of uncommon characters (e.g., "&" or "@") to make it easy to find redacted information at any point. Deferring this conversation until later in the discovery process complicates and adds expense to the production of database information, as information may have to be treated more than once to meet the protocol that is ultimately negotiated.

[17] *See, e.g.*, *Chen-Oster v. Goldman Sachs & Co.*, No. 10 Civ. 6950 (AT) (JCF), 2013 WL 3009489 (June 18, 2013) (finding information in data fields are communications subject to attorney-client privilege and denying motion to compel until plaintiff could offer evidence of a waiver).

Another database production issue that benefits from early conversation is the treatment of information that is inadvertently disclosed. Because database information is not well suited for inclusion into most, possibly not any, document review platforms, this information may not be scrutinized as closely as the discrete electronic files and email messages that make up the bulk of most ESI productions. As a result, the risk of inadvertently producing protected personally identifiable information may be higher in productions of database information than in production of other forms of ESI. Accordingly, parties are well advised to discuss protocols and consequences of producing or encountering inadvertently produced database information, including stipulation to an appropriate protective order.[18]

2. Use and Role of Consultants and Technology Partners

Discovery of database information differs in many respects from discovery of email and file-based ESI, and data collection and review of databases are the two phases of the discovery lifecycle that vary most dramatically. The technical and logistical nuances in producing and receiving information extracted from databases create many opportunities for errors in the process. Thus, responding parties and their counsel may wish to use consultants and other technology partners to assist in preserving, extracting, analyzing, and producing data from databases. Likewise, requesting parties may want to employ subject matter experts to help analyze and understand the database information received in discovery. Involving these consultants early in the litigation, at the meet-and-confer stage if not before, can save all parties significant time and money, and help prevent miscommunication and duplication of effort.

It must be noted that not all e-discovery consultants have the requisite understanding of the technical aspects of database discovery, and parties should be careful to ensure any potential consultants have the actual expertise to address and resolve the database discovery issues present for the particular situation. For example, consultants and technology partners used by the responding party should understand that standard forensic collection practices may not be applicable to large enterprise databases and that separate verification and validation procedures may be required for extracted data. Consultants for receiving parties should be familiar with ways to review extracted database information. Analyzing email messages and discrete electronic files typically involves a team (sometimes a large team) of reviewers and takes place through a document review platform. Such review and analytical tools, however, are a poor fit for the matrices of information found in tables of extracted database information. Instead, review of this information may require technically sophisticated analysts to query the data and extract the meaning of its aggregated information.

Few, if any, industry standards exist to measure the competence of database discovery experts and consultants. As always, when considering a potential technology partner, parties should consider the qualifications of the partner, the cost, and the defensibility of the solutions and processes that these experts suggest for the legal dispute.

[18] The Sedona Conference*, The Sedona Conference Commentary on Protection of Privileged ESI*, 17 SEDONA CONF. J. 95 (2016).

3. Impact of Remote Jurisdiction and Location

While beyond the scope of these Principles, it is important to understand that large enterprise-wide databases may pull data from multiple physical locations, including data stored outside the United States. Moreover, some U.S. companies make substantial use of databases that are stored entirely on computers outside the United States and are available only through remote access. Either of these situations may require parties to consider not only their respective needs in the immediate legal dispute, but also whether laws of foreign jurisdictions will complicate or even bar the use of database information outside the jurisdiction where the information is stored. Parties should discuss these issues early on to understand the impact of these logistical and legal limitations. Additional guidance may be found in The Sedona Conference Framework for Analysis of Cross-Border Discovery Conflicts,[19] published by The Sedona Conference Working Group 6 on International Electronic Information Management, Discovery and Disclosure (WG6).

B. Sedona Principle 5: Duty of Preservation

The obligation to preserve electronically stored information requires reasonable and good faith efforts to retain information that is expected to be relevant to claims or defenses in reasonably anticipated or pending litigation. However, it is unreasonable to expect parties to take every conceivable step or disproportionate steps to preserve each instance of relevant electronically stored information.[20]

Preservation of databases and database information can take place in a number of ways; the database structure and nature of the data it holds likely will suggest an appropriate procedure to ensure that potentially relevant data is not inadvertently altered or destroyed. The mere fact that a database contains some relevant information does not necessarily mean all information in the entire database must be placed under a legal hold. Database analysis typically starts with the most granular or atomic level possible—individual data fields—and uses relevance to guide the determination of whether information in that field should be preserved pursuant to a legal hold.

When preservation involves saving the results of a custom query or report outside the database, the specific query or report which was used to create the results also should be preserved. If preservation is done 'in place,' it is good practice to save both the query and report that was run, as well as a copy of the produced data.

[19] *See The Sedona Conference Framework for Analysis of Cross-Border Discovery Conflicts* (2008), https://thesedonaconference.org/download-publication?fid=486pub/67.

[20] *The Sedona Principles, supra* note 15. For additional guidance, *see* also *The Sedona Conference Commentary on Legal Holds: The Trigger & The Process*, 20 SEDONA CONF. J. 341 (2019), *The Sedona Conference Commentary on Proportionality in Electronic Discovery*, 18 SEDONA CONF. J. 141 (2017).

1. Burden of Preservation

The burden of preserving a database may be relatively modest if the system maintains all information that has been entered into it—i.e., the repository serves as a permanent archive as well as a source of current information. In such cases, while the exact state of the database may change over time due to the addition of new records and information, there is less of a risk that information that existed at the time that a preservation obligation arose will be lost. Similarly, if a company's retention policy and practice is to permanently retain in the database the ESI that is relevant to the claims and defenses in the case, preservation in place may be an acceptable way to meet the preservation obligation.

On the other hand, preserving database information may be more complicated when it is stored in a system that purges database records and information on a routine basis.[21] Just as some email servers may retain messages for short periods of time before automatically deleting them, some transactional databases also remove records after their information has become dated or is no longer required for ongoing operations. One approach taken to preserve such transactional information is to retain archival or disaster recovery media for the systems that capture and process the transactions. Unfortunately, this broad preservation approach includes not only potentially relevant data, but also all of the data on the system. In addition, storing historical data in this format can strain IT resources and disrupt business operations, as well as lead to substantial downstream costs when the database must be recreated as part of the process of restoring information from archival or disaster recovery media.[22]

In situations where a database lacks a permanent archival function or where there is no reasonable way to interrupt the usual purge or deletion cycles in order to support data preservation during the expected duration of the legal hold, preserving the relevant information stored within the database may require exporting a copy of some or all of the information to a more permanent storage medium. Tools that can accomplish this task include data export functions (either to static data tables or to an alternate database platform), special backups of the database (or of an appropriate portion

[21] Jay E. Grenig & William C. Gleisner, EDISCOVERY & DIGITAL EVIDENCE § 7.18 (200822) (finding that the scope and format of preservation as it relates to structured data is not straightforward, as the data is generally composed not only of the individual pieces of data, but also a method of interconnecting such data); *Paul v. USIS Commercial Com. Servs., Inc.*, No. 04-CV-01384, 2007 WL 2727222, slip op. at *1 (D. Colo. Sept. 17, 2007) (court declines to shift $292,000 in preservation costs after the parties failed to agree to narrow the scope of database discovery); *see also* Thomas Y. Allman, *Managing Preservation Obligations After the 2006 Federal E-Discovery Amendments*, 13 RICH. J. L. & TECH. 9, § 46 (Spring 2007) (when data is automatically and frequently overwritten, "preservation obligations can be difficult or impossible to execute.").

[22] The Federal Rules Advisory Committee noted in 2005 that "many database programs automatically create, discard, or update information" . . . and "that suspending or interrupting these features can be prohibitively expensive and burdensome." 13 RICH. J. L. & TECH. 9, § 46 (internal citations omitted).

of it), or by using built-in or third-party report writing functionality to identify, organize, and output the relevant information.[23]

2. Inventory and Default Retention Periods

Because of their complexity, databases often will require additional expertise beyond that of a legal team familiar with working with other sources of ESI, such as email messages and discrete files. In addition to understanding their databases and the information stored in them, parties should also be familiar with how databases may interact with one another and whether the information in the databases is permanent or transient—i.e., is deleted or purged from the database after a set period of time or when specific conditions are met.

Many databases are subject to update and modification as part of the normal course of business. In addition, practical business considerations may prevent a party from locking down data contained in a critical database. In such cases, it is critical that the party develop an alternative way to preserve the relevant ESI. For example, if the prices or product offerings of an online retailer is relevant to the claims or defenses of a case, and preventing changes to the underlying pricing and product databases that control the products available to customers would impose an undue burden on the retailer, the party could preserve the relevant ESI outside the database in the manner described in B.1. above. The retailer should, however, take proactive steps to preserve such data if it becomes reasonably apparent that time-sensitive information is likely to become relevant to a legal dispute. Failure to take appropriate proactive steps has led to sanctions or adverse inference instructions when potentially relevant data has been lost because a party's normal business practices for maintaining dynamic data sources led to the destruction of potentially relevant database information after a legal hold obligation accrued.[24] In such cases, responsive data can be preserved outside the database in the manner described in Section B. above.

C. Sedona Principle 6: Responsibilities of Responding Parties

Responding parties are best situated to evaluate the procedures, methodologies, and technologies appropriate for preserving and producing their own electronically stored information.[25]

1. Parties Must Understand Important Database Characteristics

At a minimum, parties participating in the discovery of database information should familiarize themselves with several basic database attributes so that they have adequate knowledge and

[23] *Id.* § 48 (proposing to battle the problem of preserving continuously changing data in a database by running and recording queries of the database at certain periods of time).

[24] *See, e.g.*, *Linnen v. A.H. Robins Co.*, No. 97-2307, 1999 WL 462015 (Mass. Super. June 16, 1999) (adverse inference jury instruction appropriate where responding party violated ex-parte order to preserve back-up tapes).

[25] *The Sedona Principles, supra* note 15.

understanding to develop reasonable procedures for preserving and producing information from these repositories.

- Functional Purpose: What is the purpose of the database system? A database may have field names that appear to indicate relevant information, but the actual information stored in the system may be completely different and irrelevant. Accounting systems, payroll, sales, and operations systems are database systems commonly found in many organizations. They may be critical to the ongoing company operations. However, some or even all of these systems may not contain relevant information. Understanding how data is used will help determine whether the database in which it resides should be subject to a litigation hold.

- System/Business Owners: Who are the primary users of a database? Who are the administrators who maintain the plumbing of a database? These two groups, which may or may not overlap, comprise the witness pool most knowledgeable about these systems. Database administrators/managers generally have the greatest knowledge of which users have access to the data and which users can add or modify information. Database users, on the other hand, can provide critical information about the nature and value of the information in the database that will identify whether the database is likely to be relevant. These users can provide invaluable substantive information, such as formatting inconsistencies, data anomalies (e.g., when a data field becomes used in a new way and old information is not the same as new information entered into the same field), and other functional limitations.

- Location: Parties should know the physical location of their databases and understand how the data is managed. Because many databases are located in remote server farms (e.g., co-location facilities) or even in different countries, it is possible that the law of more than one jurisdiction may apply to any database discovery that must take place. Database systems also may be managed by third-party vendors whose proprietary database management procedures are not necessarily known by, much less legally under the custody and control of, a party.

- Reports: Existing report templates or canned reports are a valuable and low-burden method for identifying and potentially producing database information. Canned reports are particularly helpful when only a subset of the information in a database is potentially relevant. Knowing what reports are available will help a party better understand the burden of complying with database discovery requests. For example, it may be possible to provide a requesting party with 80% of the database-stored information it seeks through a canned report, with extraction of the remaining information requiring a much greater effort. When presented with this information, the requesting party may defer the remainder of its request until it has a better sense of the actual relevance of this information to the legal dispute. Canned reports themselves can often be saved into database tables, providing a requesting party with validated, reliable information that can be used as produced or as raw data for further analysis.[26]

- Archival, Retention and Disaster Recovery Policies: Database systems frequently archive historical data that has exceeded its useful life and has no further business purpose. For example, online banking records often fall into this category; transaction records may be available for a discrete period before being archived and purged from the active database. Data that has been archived may still be accessible if required, although the burden of retrieving it is notably higher than when it was active data within the database. It is critical that parties to a potential suit know the extent to which database information is archived—and the schedules by which active and archived information is ultimately purged from a database system.

- Legacy Systems: In an infrastructure-upgrade project, it may be less expensive for an organization to start fresh with a new database system than to transfer all existing information from an old system. In such cases, the old legacy database may be maintained or archived in case its historical information is ever required. Legacy database systems are frequently associated with accounting or operations systems that were replaced, rather than upgraded. Orphaned legacy systems—databases or systems with no identifiable users, custodians, or technical support—also are common in merger or acquisition situations when the corporate information of one entity is no longer in active use. A party should be able to identify what, if any, relevant legacy database systems exist within its organization, as well as whether any relevant information in these systems was ported to a newer, more readily available format.

2. The Responding Party Ordinarily Should Determine the Best and Most Reasonable Way to Identify, Extract, and Produce Relevant Data from Databases

A responding party, with the advice of its counsel, is responsible for determining a reasonable method for identifying, preserving, extracting, and producing relevant data from databases.[27] However, just as a driver of a car may need a mechanic to help understand how the automobile's engine or on-board computer works, a party may require additional expertise to develop adequate

[26] Producing reports from databases in lieu of production of the database itself is supported by Fed. R. Civ. P. 34(b)(2)(CE)(iii): "A party need not produce the same electronically stored information in more than one form." But at least one court has held otherwise. *Margel v. E.G.L. Gem Lab Ltd.*, No. 04 CIV 1514, 2008 WL 2224288, at *5 (S.D.N.Y. May 29, 2008) ("[I]t appears that EGL-USA's only objection is that the database is redundant of the information that has already been produced. I do not find that objection to be persuasive in light of the fact that information maintained in an electronic database is necessarily in a form that is not identical to a report prepared on the basis of that data and should, therefore, ordinarily be produced."). The court in *Margel* did not cite Rule 34 for this proposition and instead cited a case that ordered a party "to produce paper and electronic copies of same documents." *Id.* (citing *Member Servs., Inc. v. Sec. Mut. Life Ins.*, No. 06-CV-1164, 2007 WL 2907520, at *1 (N.D.N.Y. Oct. 3, 2007)).

[27] *See In re Ford Motor Co.*, 345 F.3d 1315 (11th Cir. 2003) (stating that the responding party's choice to review database and produce only those relevant portions was adequate discovery response absent specific evidence to the contrary).

procedures to identify and produce database information. Normally, such expertise, whether through consultants, IT professionals, or other specialists, serves as an adjunct to the responding party's legal team. In highly disputed situations, however, courts may choose a neutral third party, such as a special master, to assist with this process.[28]

3. Parties Must Consider the Database as It Is, Not as It Could Be

Databases may be in service for extended periods of time, evolving with the needs of the organizations that created them. However, older systems may be unwieldy or inefficient when compared to current or newer database applications and installations. This can lead to frustration (by all parties) with the functionality of a given database, and claims by a responding party that certain requests for information stored in a database are unduly burdensome. Requesting parties have challenged such claims of undue burden, arguing that a responding party may not rely upon idiosyncrasies and limitations in its systems to establish burden; parties may not hide behind a unique and burdensome data management system which they created. However, absent evidence that a party has purposefully designed its data systems to thwart discovery, such challenges are not supported by Fed. R. Civ. P. 26(b)(2)(C)(iii) and its state analogs as those rules implicitly hold that the requesting party finds the producing party's database system as it is.

A number of courts have held that absent a statutory requirement to maintain data in a specific manner or in the absence of a specific preservation obligation, a company may maintain its corporate information in any manner it chooses, so long as its system is not intentionally designed to frustrate discovery.[29] As a consequence, a requesting party finds a producing party and its IT systems as they are and not as they wish them to be.[30]

[28] *Maggette v. BL Dev. Corp.*, No. 07-CV-181, 2010 WL 3522798 (N.D. Miss. Sept. 2, 2010) (inability of a party to retrieve relevant information from one or more of its databases over the course of five years, required the appointment of a special master).

[29] *See Arthur Andersen LLP v. United States,.S,* 544 U.S. 696, 704 (2005) (endorsing business practice of routine records destruction).

[30] *Jones v. Goord*, No. 95 CIV. 8026, 2002 WL 1007614, at *10 (S.D.N.Y. May 16, 2002), claim dismissed, *Jones v. Goord*, 435 F. Supp. 2d 221, 266 (S.D.N.Y. 2006). The court described the interconnected and interrelatedness of the data as follows:

> [T]he databases in question are not simply collections of lists or numbers that can be easily extracted and correlated with other numbers; rather, each of the requested databases has been constructed to support the interactions of hundreds of concurrent users rather than to support the analytical activities of a few. Consequently, the databases are integrally connected to a data system that comprises 25 separate but interdependent subsystems that each are comprised of scores of programs, tens of databases and scores of screen and report formats. There are over 3,000 programs containing a total of 1,500,000 lines of program instructions.

This lack of explicit legal obligation does not mean that an organization should not consider litigation discovery issues and potential costs when choosing or implementing a new database. However, the organization is not required to design or implement its databases around the potential for litigation. Virtually all databases include some design compromises after balancing competing business and legal needs. Ensuring that the database can conduct core-business functions in the ordinary course of business typically is a higher priority than ensuring that the database has capabilities for the identification, collection, and production of data that is potentially relevant and responsive to litigation. Such design decisions are appropriate, as long as they are not made to frustrate legitimate discovery.

Some courts have held that the self-imposed idiosyncrasies of a party's information management systems that make it challenging or costly to respond to discovery requests are not valid grounds for limiting discovery requests. In this line of cases, courts have applied the general principle that a litigant ordinarily bears the costs of collecting and producing relevant discoverable evidentiary materials, even if the litigant's discovery costs are unusually high due to the way that the responding party has chosen to organize its business records.[31] But high costs should factor into the courts' proportionality analysis, unless the party purposely designed its data systems to thwart discovery.

When analyzing production difficulties due to limitations in a database design, underlying database engine functionality, or data integrity, parties should consider a variety of data production options to see which best meets the needs of both requesting and producing parties. For example, it may be possible to extract and produce relevant data with relatively modest burden if it is bundled with some amount of non-responsive database information. In this circumstance, particularly if the responding party produces the data as it has been kept in the ordinary course of business, such a production may satisfy the responding party's obligations, so long as the burden of extracting responsive data is roughly equal for both[32]parties.

4. Direct Examination of Databases

Absent the parties' specific agreement, a requesting party is rarely granted permission to conduct a direct examination of a responding party's database to view or obtain information stored within it. As also noted in the commentary to Sedona Principle 6 above, most litigation discovery requests relate to a database's content, not how it operates. Allowing full access to a responding party's

[31] *See Static Control Components, Inc. v. Lexmark Int'l, Inc.*, No. CIV.A. 04-84, 2006 WL 897218, at *4 (E.D. Ky. Apr. 5, 2006) ("The Federal Rules do not permit Lexmark to hide behind its peculiar computer system as an excuse for not producing this information to SCC."); *In re Brand Name Prescription Drugs Antitrust Litig.*, No. 94 C 897, 1995 WL 360526, at *2 (N.D. Ill. June 15, 1995) (producing party cannot shift discovery costs to class action plaintiffs where "the costliness of the discovery procedure involved is . . . a product of the defendant's record-keeping scheme over which the [plaintiffs have] no control") (citation omitted) (alterations in original); *see also Dunn v. Midwestern Indem.*, 88 F.R.D. 191, 197-98 (S.D. Ohio 1980); *Kozlowski, PPA v. Sears, Roebuck and & Co.*, 73 F.R.D. 73, 76 (D. Mass. 1976).

[32] It should be noted that a responding party is never obligated to produce non-relevant information. *See* Section III. Comment 1.F., *infra.*

database makes it difficult, if not impossible, to prevent the requesting party from accessing irrelevant or privileged information; all data fields in all database records are theoretically accessible. Direct access to a proprietary database by a non-employee also may compromise the validation of the data in the database, reducing the database's reliability for both business and legal situations.

All this said, in certain civil litigation matters, responding parties have, in fact, invited requesting parties to access one or more of their database systems as an alternative to producing relevant information by exporting it or by cloning the database.[33] Typically, the databases in these cases contain no personally identifiable information; for example, a database of manufacturing information. Typically, too, the requesting party is often supervised, either by a responding party representative, or by a neutral third party. In some cases, the requesting party has agreed not to directly access the system, instead directing an employee of the responding party to enter queries and otherwise manipulate the system. Finally, the requesting party usually must sign stringent confidentiality agreements to prevent the inadvertent disclosure of any proprietary information (relevant or irrelevant) that the requesting party may see when accessing the database.

Direct access to a party's database systems is disfavored and has been granted over objection only in extraordinary circumstances. In re Ford Motor Co.[34] is a rare case that discusses this issue directly in the context of database discovery.[35] The plaintiff had requested direct access to Ford's databases to conduct queries for claims related to defective seatbelts. However, the court held that "Rule 34(a) does not grant unrestricted, direct access to a respondent's database compilations. Instead, Rule 34(a) allows a requesting party to inspect and to copy the product—whether it be a document, disk, or other device—resulting from the respondent's translation of the data into a reasonably usable form."[36] The court further explained that Rule 34(a) contemplates that the responding party will search its own records directly to produce the records, not that the requesting party directly searches the data itself.[37] The court held that while some kind of direct access might be permissible in certain cases, this case was not one of them because the plaintiff's request was too broad in scope and because the district court made no findings that Ford had failed to comply with discovery requests.[38]

[33] *See, e.g.*, *OpenTV v. Liberate Techs.*, 219 F.R.D. 474, 475 (N.D. Cal. 2003) (in software patent infringement suit, responding party offers to grant requesting party access to its extensive source code database, but court orders parties to share cost of data extraction).

[34] *In re Ford Motor Co.*, 345 F.3d 1315, supra.

[35] *Id.*

[36] *Id.* at 1316-1317.

[37] *Id.* at 1317.

[38] *Id. See also Cummings v. Gen. Motors Corp.*, No. CIV.00-1562-W, 2002 WL 32713320 (W.D. Okla. June 18, 2002), *aff'd*, 365 F.3d 944 (10th Cir. 2004), *as modified on denial of reh'g* (June 2, 2004); *Butler v. Kmart Corp.*, No. 05-CV-257, 2007 WL 2406982, at *3 (N.D. Miss. Aug. 20, 2007); *but see Qualcomm, Inc. v. Broadcom Corp.*, No. 05CV1392-B, 2007 WL 935617 (S.D. Cal. Mar. 13, 2007) (to resolve discovery dispute over search terms applied to a proprietary Oracle database, the Court ordered the responding party to provide the requesting party access to a full version of the

5. Documentation and Validation of Database Collections

When extracting data from databases for production, it is important to document, test, and validate the procedures that are used. The internal documentation may include the steps taken, who performed the work, when it was performed, and what was the result of each step. Well-documented data collection and production procedures enable a responding party to demonstrate its good-faith efforts to accurately export and produce database information. The same documentation also makes it possible to respond to any allegations of over- or under-collection of database information.

6. Features and Limitations of Technology and Tools that can be Applied to Databases to Identify and Extract Relevant Information

Databases differ in the types of functions that are incorporated into them. For example, some databases support open-ended, free-form text fields; others impose much shorter character length limitations on their data fields. All databases offer search query functionality, but some database engines support deeper search functionality than others. Still other database engines may offer powerful search features but may index only the first several hundred characters in a data field, making standard search queries unreliable when applied to long, free-form data fields.

Responding parties have an obligation to understand the features—and shortcomings—of the database engines that power their information repositories. Understanding this technology is separate from the data content or system usage knowledge required to explain the significance of database field names or how information was entered into the structure. Indeed, different individuals within an organization typically have one, but not both, of these distinct bodies of knowledge about its databases.

Understanding the limitations of a database also requires an understanding of which external utilities—if any—can be used to add functionality to a database. For example, the software that powers many enterprise-class databases may be relatively limited in the ways that it can format information into reports. Instead, these database engines allow close integration with third-party report generation tools. Because of the variety of ways that a database can store its information, however, not all reporting or other enhanced functionality tools will work with all databases or database systems.

A responding party may not be able to meet its database discovery obligations without solid knowledge of these tools and their potential application to the party's relevant databases. Without this understanding, it is difficult for a responding party to fully understand, much less articulate, the burden that a given discovery request imposes on it. Moreover, a lack of this knowledge greatly limits a party's ability to have comprehensive, frank discussions about database discovery.

database, including the same search capability and client tools used by producing party engineers, along with a one-hour live training tutorial and written instructions on how to use the search tools).

D. Sedona Principle 12: Form of Production and Metadata

The production of electronically stored information should be made in the form or forms in which it is ordinarily maintained or that is reasonably usable given the nature of the electronically stored information and the proportional needs of the case.[39]

1. Mismatch of Native Format to Most Database Productions

Rule 34(b)(ii) and its state equivalents mandate that a responding party must produce ESI in either the form or forms in which it is ordinarily maintained (sometimes called "native format") or in a reasonably useable form or forms.[40] However, native format may not have as clear a meaning in a database context as it does for other forms of ESI.[41] In fact, in many cases, a truly native format production of database information is less usable to a requesting party than an alternative production format.

The 2014 Database Principles focused on Relational Database Management Systems (RDBMS) which are comprised of fields within tables, stored procedures, functions, and an application or engine which allows for queries. In the ensuing years, several additional database technologies have been implemented with characteristics that differ from RDBMS. Examples include key-value stores, document stores, column-oriented databases, and graph databases, each of which has its own unique structure, format, and engine. As new database technologies come online and evolve, Native Format becomes an even more complex issue than for RDBMS. The four types identified above are "Schemaless" types, which do not have a formal data model and can evolve organically through use. Complicating matters further are multi-model systems which rely on more than one type and may integrate an RDBMS for portions of the database.

Database engines may compact the information they store and index to reduce storage requirements and speed information retrieval. Each database engine uses a different proprietary format for the data files that make up the components the database uses to properly function. For example,

[39] *The Sedona Principles, supra* note 15.

[40] In several instances, courts have held that databases should be produced in native format. *See, e.g.*, *In re NVMS, LLC*, No. 308-01901, 2008 WL 4488963, at *1 (Bankr. M.D. Tenn. Mar. 21, 2008); *Covad Commc'ns Co. v. Revonet, Inc.*, 258 F.R.D. 5 (D.D.C. 2009). *Compare with Coquina Investments v. Rothstein*, No. 10-60786-Civ., 2012 WL 3202273 (S.D. Fla. Aug. 3, 2012), *aff'd sub nom. Coquina Invs. v. TD Bank, N.A.*, 760 F.3d 1300 (11th Cir. 2014) (finding that counsel should have produced a requested document in native format to preserve its original qualities but declining to award sanctions) *and In re Facebook PPC Advert. Litig.*, No. C09-03043 JF (HRL), 2011 WL 1324516, (N.D. Cal. Apr. 6, 2011) (ordering parties to meet and confer regarding an alternative to producing a proprietary database storage format when a PDF printout of the database did not show data fields, hence the database was not produced as it appears).

[41] *See, e.g.*, *Bob Barker Co. v. Ferguson Safety Prods., Inc.*, No. C 04 04813, 2006 WL 648674, at *4 (N.D. Cal. Mar. 9, 2006) (declining to order production of financial services database responsive to discovery request because "it is unclear how a party could go about producing 'a database,' which ordinarily is a dynamic collection of data that changes over time").

Microsoft Access often folds all database information into a single .MDB format file. A Microsoft SQL Server database, on the other hand, is composed of several types of files, including primary files (.MDF), secondary files (.NDF), and transaction logs (.LDF). Other database engines use different structures and file types, and few, if any, can read or process information stored in a different database engine's format.

A true native production of database information provides a copy of a database that can be used only by someone possessing a licensed copy of the correct version of the database engine software, and in the case of NOSql databases, may require additional infrastructure for components of the software. Depending on the nature and age of the original database, such a license may be difficult for a requesting party to obtain, if not practically impossible. Further, the complexity and interrelation of these systems makes Native Format production a misnomer as data may be federated across multiple systems for aggregation, analysis, or storage purposes. An additional disadvantage of producing a database in its native format is that internal tracking may be difficult or impossible to turn off. Stated another way, this means that merely opening a database may alter some of its validation values such that the authenticity (and thus admissibility) of the database can no longer be established at the native file level.

While a true native production of database information may not be feasible or desirable, some metadata—in the generic sense of the term, information about information—is necessary for the production to make sense. This is a distinguishing feature of database information. As one court discussing Sedona Principle 12 put it, "while metadata may add little to one's comprehension of a word processing document, it is often critical to understanding a database application."[42] And the same court, comparing different form-of-production options, noted "[o]ne marked disadvantage of [TIFF or PDF] is that the production involves significant costs; it also does not work well for spreadsheets and databases."[43] In the context of NOSql databases, this issue is even more acute—Metadata may be embedded within the text of a document and leveraged by the engine to create documents and relationships between data when rendering what appear to be documents.

If a requesting party receives a native-file database production from an RDBMS, the native production should be accompanied by a production of database information in the form of generic data interchange files such as text delimited files data, interchange files such as text delimited files that can be read by many different types of databases or other software applications. Such data interchange files should include the fielded data that has been exported, so the requesting party can use the load files to map each information field into a database structure of its own design. Field types should be provided for mapping and field definitions provided upon request to help ensure data is appropriately understood.

[42] Aguilar v. Immigr. & Customs Enf't Div. of U.S. Dep't of Homeland Sec., 255 F.R.D. 350, 354 (S.D.N.Y. 2008).

[43] *Id.* at 356.

It is important that the producing party's counsel or consultants (see Section I.E, supra) understand the structure of the database prior to any meet-and-confer on the subject, and whether a native production is viable.

2. Use of Standard Reports to Produce Database Information

As addressed in I.E., I.F., and II.B.1, supra, some databases include ways for business users to view or print out multiple data fields, organized in a useful manner. The simplest database reports might present columns of information in a simple table format. More complicated reports may combine content from multiple fields, perform mathematical calculations and present them, or include graphs derived from underlying database information. Database reports may be static—that is, an unchanging view of certain data that have been selected by query, or they may be more interactive, permitting users to change the scope, focus, and perspective of the database. Generally speaking, most existing reports that are used in day-to-day business are "pre-validated," meaning that accuracy of their data aggregation has been tested and demonstrated. Standard reports, also known as "canned" reports, should be contrasted with custom reports, where users (or database administrators) select report content based on individual or changing needs. Because these reports are created on the fly by database users, it is more possible for these information views to include errors, such as mismatches between field name and displayed field contents or mathematical errors.

Standard reports have both advantages and disadvantages as a production format for database information. Because these report templates already exist and have been pre-validated for accuracy, it is generally faster and cheaper to use these reports than to create custom views and information extracts. However, standardized reports may not collect all potentially responsive or relevant data in the database, and they may not produce it in the specific format that has been requested. Thus, standardized reports may be a low-burden way to make a partial production of requested database information, but they may not provide the most complete solution.

Additionally, to the extent reports exist, a preference for them may be appropriate as they are how a business views its data in the ordinary course of business. The data is not stored within the report, rather it is stored in tables, key-value pairs, or documents within the database, but it may be how data is used in the ordinary course of business.

If a standardized report is missing crucial data or provides information in a way that cannot be processed using reasonable efforts by the requesting party, a different production format may be more suitable. On the other hand, if the standardized report captures all the significant data and omits only marginally relevant information, it may be more appropriate to produce database information in a standardized report than to invest time and money into creating a custom report that provides absolutely all of the database information that has been requested. To assess the completeness of a given report, it may be useful to share the template structure absent data for discussion during an early meet-and-confer.

Requesting parties must consider databases as they are (as discussed) and should consider their intended use cases for the data they are requesting.

> **Illustration i**: Company A sells widgets and has a system to track inventory and sales. The requesting party intends to show the long-term sales numbers do not reflect the current market value of goods at the time that they were sold. In order to do this, the requesting party asks for all sales records for the period covered by the litigation. Company A's lawyers offer canned reports which provide quarterly graphs of sales, which is the way corporate executives view the data. In this instance the reports would not meet the need of the requesting party despite capturing the data and being utilized in that way by a specific business unit. This would likely not be acceptable unless the underlying data was prohibitively difficult to obtain. These questions should be carefully considered by requesting parties and addressed as early as possible, optimally in the early meet-and-confer.

3. Use of Fielded Tables to Produce Database Information

A common way to produce database information for an RDBMS is through tables (i.e., rows and columns) of information, where each row represents a database record and each column represents a single data field. This is not the case in other database technologies. For example, a columnar database, which at first looks the same as an RDBMS, turns this on its head and stores all data in columns and rows which do not have field types, searches only within columns rather than entire rows, and associates each field within a record with a row number for analysis. Many database engines, even those that do not have sophisticated reporting functionality, support exporting database information into either text delimited files or fielded tables. Similarly, many database engines can import data interchange files and separate out each field of information for subsequent analysis.

Text delimited files are closely related to, if not often virtually the same as, database load files; they are generically formatted sets of fielded information. Delimited files, however, may not be able to completely show the relationships found in multi-table relational databases. For example, in a banking database, a single customer may have both individual bank accounts and a shared bank account with one or more co-owners. Typically, these relationships are tracked in a multiple-table relational database, where each bank customer can be related to multiple bank accounts, and each bank account can be related to one or more customers. If this information must be consolidated in a single table, preserving these one-to-many relationships may require that information be repeated so that full information can be displayed in each view of the information. Denormalizing the data in this way (i.e., transforming it into a different format from the way in which it is stored in the ordinary course of business) is a relatively common and often acceptable data production practice, even though restoring this information into multiple relational tables to recreate the original types of relationships may not be a straightforward process, depending on the data relationships that are required. More recent systems may store this data in a GraphDB structure which maintains each data point as a node with edges between nodes identifying their relationships. In that case, the one-to-many relationship model is not implemented in the same way, and the format of production will be driven by the technology involved.

For example, the parties could clarify whether the requesting party would prefer to see the results of a query or report that links the data elements together, or to have exports of the responsive data from separate tables, in the case of an RDBMS, or selected node-edge structure for a GraphDB, and import the files into their own system in order to run their own queries.

III. THE SEDONA CONFERENCE PRINCIPLES FOR THE PRESERVATION AND PRODUCTION OF DATABASES AND DATABASE INFORMATION (THE "SEDONA DATABASE PRINCIPLES")

While The Sedona Principles cover the preservation and production of ESI in general, and include useful guidance for the discovery of databases and database information in particular, the complex and evolving nature of database discovery calls for a more in-depth examination of the issues that are unique to databases and the information found in them.[44] Because of the structural complexity and volume of database information, database preservation, collection and production often involves relatively greater costs and burdens than those associated with the production of unstructured media. Defining a reasonable scope of database discovery requires all parties to understand the purpose for which the information is sought, the components and respective relevance of the data at issue, the workings of the technology that stores and manipulates the data, and the processes to ensure that the data produced is what it purports to be. To that end, the following six Sedona Database Principles are intended to inform and facilitate discussions regarding assessments of relevance, potential costs and burdens, and methods for validating results that necessarily must occur between parties that are involved in database production.

1. Scope of Discovery

Absent a specific showing of need, a requesting party is entitled only to database fields that contain relevant information, and give context to such information, and not to the entire database in which the information resides or the underlying database application or database engine.

Comment 1.A: "New" Database Reports Are Within the Scope of Discovery

Generally, a party cannot be required to produce a document that does not exist.[45] This general principle arises from Federal Rules of Civil Procedure 34 and 45, which establish that the scope of discovery extends only to those documents in the responding party's "possession, custody, or control."

Thus, in theory, a responding party could argue that it is not required to query a database and create new reports to respond to a discovery request. Under that argument, the pre-existing database is the document subject to discovery, but new reports generated from it are not.

Whatever the theoretical merit of that argument, though, courts have generally not accepted it. Instead, courts have held that "'requiring a party to query an existing database to produce reports for

[44] The authors also wish to call the readers' attention to The Sedona Conference, *Commentary on Proportionality in Electronic Discovery*, 18 SEDONA CONF. J. 155 141 (2017).

[45] *See*, *e.g.*, *Manning v. General Motors*, 247 F.R.D. 646, 652 (D. Kan. 2007).

opposing parties' does not equate to requiring the creation of a new document."[46] "[C]ourts "regularly require parties to produce reports from dynamic databases, holding that the technical burden . . . of creating a new dataset for litigation does not excuse production."[47] Some cases require producing parties to create "a custom-created query or report."[48] The narrow exception to this holding appears to be a situation where the requested information does not actually exist in the database that would be queried.[49]

These court cases make eminent practical sense. If a responding party could not be compelled to prepare new queries from a database, the requesting party would either be unable to discover information contained in the database or would be obliged to request the database in its entirety. The former would be inconsistent with the generally broad scope of discovery, while the latter would require the responding party to disclose voluminous non-responsive (and potentially privileged or sensitive) information.

Comment 1.B: Database Relevance Must Be Analyzed on a Granular Level

Databases are often very large collections of disparate information. Although situations can exist when an entire database and its information are relevant to a legal dispute, often only a portion of a database is relevant.[50]

The process of determining which database information is relevant is performed at several levels. First, depending on the nature of the dispute, many database records will likely not contain relevant information. These normally would be excluded from production through use of search queries. Second, however, even within records that contain potentially relevant information, not all the

[46] *McGlone v. Centrus Energy Corp.*, No. 2:19-cv-2196, 2020 WL 4462305, at *3 (S.D. Ohio Aug. 4, 2020) (quoting *Mervyn v. Atlas Van Lines, Inc.*, No. 13 C 3587, 2015 WL 12826474, at *6 (N.D. Ill. Oct. 23, 2015)); *see also Dept. of Finance v. AT&T Inc.*, 239 A.3d 541, 574 (Del. Ch. 2020) ("Querying a database and extracting or exporting information does not constitute the creation of a new document. It is how a party accesses an electronic records-keeping system in the ordinary course of business.").

[47] Loc. 3621, EMS Officers Union, DC-37, AFSCME, AFL-CIO v. City of New York, No. 18CV4476LJLJW, 2024 WL 1856302, at *2 (S.D.N.Y. Apr. 26, 2024) (internal citation and quotation omitted).

[48] *Id.*

[49] *See Adams v. Target Corp.*, No. 1:21-cv-3352, 2021 WL 3620280, at *3 (N.D. Ill. Aug. 16, 2021) (citing *Conn. Fair Hous. Ctr. v. CoreLogic Prop. Sols.*, No. 3:18-CV-705 (VLB), 2020 WL 401776 (D. Conn. Jan. 24, 2020)).

[50] *See In re Lowe's Cos., Inc.*, 134 S.W.3d 876 (Tex. App. 2004) (granting mandamus and vacating trial court's order for retail chain to produce database for query by requesting party without any limitations as to time, location, or subject matter); *Ex parte Wal-Mart, Inc.*, 809 So.2d 818 (Ala. 2001) (mandamus granted in part to restrict requesting party's access to retail chain's incident reporting database to similar incidents only). *See also Barnes v. District of Columbia*, 289 F.R.D. 1 (D.D.C., Sept. 28, 2012) (granting motion to compel search algorithm because a query used to search a database and generate reports is a "writing" subject to production but denying request to access entire database as overbroad).

data fields that comprise the record may be relevant.[51] Identifying and extracting database information in response to discovery requests requires both levels of analysis.

The process may be complicated further by the differing views available to users based upon different levels of database security access. A database record in a database application that is viewed on the screen by a typical end user generally is created from information stored on multiple data tables, and only database administrators may be able to see the raw data as it is stored in database tables and subtables. Unfortunately, many database discovery requests combine requests for both database records and database tables as if they were separate and mutually exclusive repositories of information. Depending on the technological sophistication of the party representatives managing this discovery, such terminology mixing can further complicate the process of reaching consensus on the logistics of these discovery requests.

Other times, the way that database fields are organized into columns, rows, and tables may simplify conversations about the scope of production. Depending on the facts in a dispute, entire tables of database information may not be relevant and may not be required to be preserved or produced. Conversely, other data tables may contain fields of important information that require special treatment. To the extent that data is rolled off or updated in an active database, a database administrator may need to implement preservation measures for specific tables to reduce the risk of inadvertently destroying potentially relevant information.

> **Illustration i**: In litigation involving a car manufacturer and the various warranties provided to consumers, plaintiffs request documents to identify the customers of certain models of cars, the cars they purchased, and the warranties they purchased. The defendant's database that retains this relevant data also contains non-relevant information, including dealership, the salesperson, and the commission the salesperson received on selling the car. This non-relevant information is stored in the same rows and tables as the responsive, relevant information. The information in these data fields is not relevant to the dispute, and the data fields do not need to be produced. Furthermore, even though both the relevant and nonrelevant information might appear in a standard view of the customer's database record, the responding party should not be obligated to produce the nonrelevant information even if the requesting party asked for all documents related to customers of the certain car models.
>
> **Illustration ii**: In a breach of contract litigation between two companies where the amount paid by one to the other is in dispute, the defendant's accounts-payable database could contain potentially relevant information regarding payments by the defendant to the plaintiff. However, absent a persuasive argument to the contrary, the data records (i.e., rows)

[51] *See, e.g.*, *Bob Barker Co. v. Ferguson Safety Prods.*, 2006 WL 648674, at *4 (N.D. Cal. Mar. 2006) (declining to order production of financial services database responsive to discovery request because "it is unclear how a party could go about producing 'a database,' which ordinarily is a dynamic collection of data that changes over time").

regarding payments to other companies for unrelated transactions are not relevant and need not be produced.[52]

Illustration iii: In the same breach of contract litigation, not every data field (i.e., column) displayed in a record that contains relevant information in the accounts payable database is necessarily relevant and within the scope of discovery. For example, the payee, amount, date, check number, approver and comments data fields (and their relationship to each other) may all be relevant, but other data fields in the record may not be relevant (e.g., unique record ID, tax ID, etc. . . .). Id.

Comment 1.C: Parties Must Determine the Relevance of Individual Data Fields Within a Database

When reviewing the relevance of data fields, parties need to carefully examine the relationship between relevant data fields and other fields (or rows, or columns, or tables). This relationship can make otherwise irrelevant data relevant because of its link or connection to relevant information. While it is possible that a single piece of relevant data within a record or table may transform otherwise irrelevant data within the same record or table into relevant data because of their relation to each other, such a logical connection is by no means automatic.

A responding party that finds relevant information in a portion of a database should reasonably consider the entire database to determine if other portions are relevant to the dispute. A party that unilaterally examines its own databases to determine what fields are relevant or irrelevant should, as a matter of best practice, act conservatively to avoid inadvertently excluding relevant data. Generally speaking, the cost of performing this analysis a second time, plus the downstream acts of extracting and processing this information a second time, is far more than the cost of identifying, extracting, processing, or producing slightly more data during a single pass.

Analysis regarding the relevance of information contained in individual cells is not unlike that pertaining to information contained in various types of metadata. In addressing the relevance of metadata associated with various forms of ESI in Aguilar v. Immigration & Customs Enforcement Div.,[53] the court drew from Principle 12 of The Sedona Principles[54] noting that "the two 'primary considerations' should be the need for and the probative value of the metadata, and the extent to which the metadata will 'enhance the functional utility of the electronic information.'" A parallel approach should be used to determine relevance of data fields (i.e., to what extent is the particular field data or its relationship to other fields essential to understanding the information sought; does such field-level data enhance the utility of the records). The Aguilar court noted that, "[a]s a general rule

[52] *See Ex parte Wal-Mart, Inc.*, 809 So.2d 818.

[53] *Aguilar,* 255 F.R.D. at 356.

[54] *Supra* note 15.

of thumb, the more interactive the application, the more important the metadata is to understanding the application's output."[55]

If the data fields themselves are not privileged or determined to be trade secret, metadata-type database field information can be analyzed in several ways for relevance. However, in Aguilar, because the data was sensitive, the court suggested a quick demonstration to the plaintiffs of database functionality using dummy data stored in an otherwise identical database structure.[56] This approach could be used as an exploratory tool with a requesting party or with fact experts to gain an understanding of the overall output from the database if the parties cannot agree on the fields or cells that may be relevant to make meaningful use of the data, or if the producing party lacks this level of understanding of its database systems.

Comment 1.D: Database Relevance Is Measured by its Data, not the Application

Under normal circumstances, a database is relevant to a legal dispute because of the database information stored within the tables or files, not the database application or database engine.[57] Unless there is a unique relationship between the database information and the mechanism that manages or displays that data (which can happen in some older or proprietary database systems), the software components of the database application and engine are unlikely to have any relevance to the discovery request, and should be considered presumptively nonresponsive.

Proactively focusing database discovery requests on the data component of the system greatly simplifies the process of responding to these requests while rarely sacrificing full disclosure. Moreover, because database systems are configured for specific hardware and software environments, the effort to recreate these environments is vastly more expensive and complex than providing the data files in a format that can be loaded into whatever database systems are available to the requesting party.

Fortunately, most database information can be produced easily in a generic format that does not require a specific database engine or application to be read or analyzed. Depending on the requesting party's needs, a data file in a common form such as Microsoft Access or Excel can be produced and allow the database information to be reasonably usable by the receiving party. Additionally, limiting database discovery to the database information which can be produced in an alternative reasonably usable tabular form obviates the need to negotiate the terms of a protective order or other limited use agreement with the non-party proprietor of the database software, cloud computing service provider, or computing platform provider.

[55] *Aguilar,* 255 F.R.D. at 354-55, (quoting *Williams v. Sprint/United Mgmt. Co.*, 230 F.R.D. 640, 647 (D. Kan. 2005).

[56] *Id.* at 363.

[57] *See* Section I.B., *supra,* for definitions.

Comment 1.E: Circumstances When a Database Application May Be Relevant

In certain circumstances, the database application, structure, or even the database engine, may not only be relevant, but also essential to providing a complete response to a discovery request, for example, when the software itself either: (a) contains information relevant to the matter not otherwise stored in the database storage file; or (b) the software is the focus of one or more claims of the litigation.

> **Illustration i**: Acme Corp. has programmed its financial system to provide a limited number of choices when categorizing financial transactions. The universe of possible choices, rather than the history of actual choices, has become an issue in litigation. Acme Corp. has been asked to produce the software application that contains the programming of these possible choices. It is clear that the database storage file will not contain this information. The parties should determine whether the production of the software is the best or only way to establish this information.
>
> **Illustration ii**: It has been alleged that, for a two-year time period, Mortgage Broker Company's (MBC) software incorrectly calculated monthly mortgage payments. MBC has been asked to produce the historical transactions, as well as the software code, that it used to calculate those transactions. It is clear that the database storage file does not contain those calculations. The parties should determine if the production of the software is the best or only way to provide information regarding the underlying algorithms used by MBC's software.
>
> In some cases, it may be more valuable to understand the database application than to receive the underlying transactional data. This situation occurs most often when one set of data (dataset A) is acted upon by a software tool to then produce a second set of data (dataset B). For such discovery requests, it may be more effective to understand the processes that transform dataset A into dataset B, rather than to simply receive dataset B, or dataset A.
>
> **Illustration iii**: Franchise Food Co. tracks employee time and attendance via its point-of-sale system (POS, i.e., the cash registers). The POS terminals record the time that cashiers signed into and out of the system. In wage and labor litigation, it has been asked to produce all POS time entries and to produce all payroll-system records. While the presumption is that both would be produced, it may be equally sufficient or even preferable to produce the POS time entries and the software that creates the payroll system records from the POS data, rather than the static payroll-system records.

Comment 1.F: Value of Information About the Database System

In addition to disputes about the relevance of database information, or the database applications or engines themselves, requesting and responding parties often disagree about the relevance of the database system information, such as the database's schematics or the underlying technical information. In these disputes, the requesting party is seeking information that is not directly at the heart of the dispute, but nevertheless may help the requesting party better understand the information that it is receiving and any limits in its accuracy or functionality. Understanding the context, origin and normal business use of ESI in a production may be helpful for the requesting party to make effective use of the data received. So while these kinds of requests may be considered discovery on discovery (also known as meta discovery), courts have been inclined to permit them.[58]

Accordingly, in appropriate circumstances, a responding party may be required to produce the database system information that is reasonably needed by the requesting party to obtain a basic requisite understanding of the structure, content, and format of the data being produced, including relevant field names and values, the relational connections between data fields and tables, and the extent to which data fields are automatically populated by the system. In some circumstances, the scope of this system information may be expanded to include not just information about the specific data being produced, but also information about where the produced data originated from within a larger environment that may include multiple database servers, internal or external databases, and other related ESI. The production of such database system information might also include dependencies of the produced data on other data sources, uses of the produced data within the system or overall environment, and relationships of the produced ESI to other data within the system or the overall environment.

> **Illustration iv**: In Illustration iii above, where Franchise Food Co. could have produced the POS time entries and the software that creates the payroll system records from the POS data, in lieu of the static payroll system records, Franchise Food alternatively may have been able to produce the POS time entries and, if available and reasonably accessible, background system technical information about the software that creates the payroll system records from the POS data.

Database system information may be presented in many ways. Sometimes, tabular or graphical depictions of a complex data system, as can be found in entity relationship diagrams or data flow diagrams, may be both most helpful and least burdensome for a responding party to provide. Other times, it may be necessary to depose a witness with technical understanding of the system from which database information has been produced. Requesting parties should understand that there is rarely, if ever, a single, comprehensive source of the system information that they may request, and that a responding party has a burden of varying degree in collecting such information for production.

An additional consideration is that information produced from databases is rarely an exact copy of the data tables and database structure. Rather, the database information being produced is most often a subset of the sometimes-substantial information that is stored in a larger database. In fact, this is often preferable.[59] Depending on the issues in the case, it may be appropriate for a requesting

[58] *See, e.g.*, *Winfield v. City of New York*, No. 15CV05236LTSKHP, 2018 WL 840085, at *6 (S.D.N.Y. Feb. 12, 2018)

[59] *See* Section I.E. Instead of being exact duplicates of existing data tables, the information is typically compiled from multiple tables (a "denormalized view") and includes fewer than all fields or records stored in a given table (a "selective view")—thus providing a variant but useful view of the data stored in the system.

party to receive a description of the extraction and transformation process, including how the produced information was organized in the original database.

A final issue regarding the production of database system information is the extent to which database or system documentation is encompassed by a request for substantive information stored in a database. Organizations do not permanently retain all database system documentation they ever create, use, or reference. Absent explicit notice from opposing counsel or other extraordinary factors, a responding party should not be automatically obligated to preserve all supporting database system documentation merely because the party has reason to believe that some ESI stored in the database may be potentially relevant to a party's claims or defenses in a current or reasonably foreseeable litigation. Commercial documentation, in particular, is usually available from a variety of sources, including third parties. More careful analysis may be required in situations involving custom-written documentation, such as internal guides or references. For such materials, responding parties should consider the nature of the documentation, as well as the degree of unique insight that this material provides into relevant database information.

Comment 1.G: Appropriate Circumstances for Producing Additional Nonrelevant Database Information

While a responding party is not obligated to produce more data from or about a database than is relevant to the dispute, in some circumstances it may be easier, less expensive, and less burdensome to produce a larger slice of the database content or even the entire database. For example, business users of the database may have a canned report that compiles all requested information, plus some additional data fields. Producing this report is likely faster and much less expensive than designing a custom query and collecting the same database information through a custom data export utility. Thus, while a responding party is never obligated to produce additional irrelevant information (and may have reasons unrelated to litigation not to do so), a responding party may produce additional nonresponsive information, so long as the responding party is not doing so for any improper purpose, such as attempting to make relevant information more difficult to extract or understand.

2. Accessibility and Proportionality

Due to differences in the way that information is stored or programmed into a database, not all information in a database may be equally accessible, and parties should therefore apply proportionality to each component of a database to determine the marginal value of the information to the litigation and the marginal cost of collecting and producing it.

Comment 2.A: Technical Challenges to Accessibility

Information from and about databases is subject to the same rules and limitations as all other information disclosures in civil litigation, and in ordinary circumstances, information that cannot reasonably be extracted using tools that are readily available in the normal course of business of the responding party need not be produced absent good cause and potential cost shifting.[60] Whether specific requested information within a database is reasonably accessible within the context of a specific legal dispute is a deeply fact-specific inquiry that must be analyzed, like questions concerning other discoverable material, under the proportionality provisions of Rule 26 and its state analogs.[61]

It is important to recognize the technical limitations that affect levels of accessibility, and a requesting party should never assume that all information in a database—or even all information visible to average database users—is equally produced. Instead, once a responding party has demonstrated why certain database information or elements are more difficult to produce than others, the parties should consider whether the value of the information is worth additional burden and cost. As with other discoverable information, the parties should consider the availability of the same information in a reasonably usable form from an alternate source (e.g., printed instruction manuals, printed database reports) and whether the importance of the requested information is proportional to the additional burden or cost that would be required to extract it from the database in which it resides.[62]

Comment 2.B: Factors for Assessing the Burden or Cost of Preserving, Collecting or Producing Database Information

Several factors may be considered in accordance with Rule 26(b)(2)(B) and Rule 26(b)(2)(C) to determine if database information may be considered "not reasonably accessible because of undue

[60] Rule 26(b)(2)(B) places specific limitation on the production of ESI. "A party need not provide discovery of [ESI] from sources that the party identifies as not reasonably accessible because of undue burden or cost." *Id.* Additionally, a court on motion or on its own, must limit the scope of discovery if the discovery sought is unreasonably cumulative or duplicative, can be obtained from a more convenient source, could have been previously obtained by the party seeking the discovery or the burden or expense of the proposed discovery outweighs its likely benefit. Rule 26(b)(2). *See also The Sedona Conference Commentary on Preservation, Management and Identification of Sources of Information that are Not Reasonably Accessible,* 10 SEDONA CONF. J. 281 (2009), and *The Sedona Conference Commentary on Proportionality in Electronic Discovery,* 14 18 SEDONA CONF. J. 155 141 (2017).

[61] *OpenTV v. Liberate Tech.*, 219 F.R.D. 474 (N.D. Cal. 2003) (court applies *Zubulake* factors to determine reasonable accessibility of source code database and allocation of data extraction costs); *Best Buy Stores, L.P. v. Developers Diversified Realty Corp.*, 247 F.R.D. 567 (D. Minn. 2007) (discovery of database denied when information sought was no longer in a searchable format, and database would have to be restored from original sources at a cost of at least $124,000 with a monthly storage cost of $27,823).

[62] *See Superior Prod. P'ship d/b/a/ PBSI v. Gordon Auto Body Parts Co., Ltd.*, 2008 WL 5111184 (S.D. Ohio Dec. 2, 2008) (where plaintiff requested production of large volume of relevant documents and where deposition witness indicated that the information would not be easily retrieved from defendant's electronic database, court recognized potential burden to defendant and ordered production of sampling of documents to allow for determination of the need to produce the rest).

burden or cost" or is disproportionate for purposes of preservation[63] or production.[64] Additionally, parties should understand that certain inherent limitations may exist impacting the production of database information.

- The extent of the ability to search on database fields: The ability to search fields depends on the way a particular database system has been designed and the sophistication of its search engines. For example, many databases contain one or more free-form text comments fields that may be visible when a database record is viewed on screen. However, to optimize performance, only the more critical, defined-format fields may be indexed and searchable, with the comments fields available only once the associated record has been located. Limiting the fields that are indexed allows databases to hold large volumes of information without compromising system performance. Third-party query/report generation tools are commonly used to supplement such limitations, however, these tools are not perfect solutions. It should be noted that searching or creating indices on unindexed fields can impose a significant burden on an operational system.

- The extent to which information may be stored outside tables: Not all information stored in a database is held in tables; it may be stored in several places. For example, to facilitate speedy and consistent data entry, a database may include predefined values for certain fields, i.e., drop down or "lookup" tables, which may be hardcoded into the database application software itself and not stored in any searchable database fields or tables. Further, earlier entries in a lookup table may not have been retained when a table or the database itself was updated, making it functionally impossible to retrieve this system information without substantial effort and expense. Therefore, a request for production seeking all values from which an employee could have chosen while engaged in data entry might sound simple on its face, but responding to this request may be extremely difficult. Likewise, certain reports may be available within a system only as screen views and not easily converted to a printable or exportable format.

- The capability for exporting data: Because information may be visible to a user does not necessarily mean that it is practically capable of being produced. For instance, individual-rights restrictions on viewing and exporting certain fields or the character of the fields

[63] Before even turning to the question of the burden and expense of *producing* information from a database, the party in possession of the database must weigh the burden and cost of *preserving* the database information (both its structure and its contents, the preservation of which are not always accomplished through the same means), against the likely importance of the information in resolving the issues in the case. *See,* Rule 26(b)(2)(C)(iii). *See* the discussion of Sedona Principle 5 *supra* at II.B. For additional guidance, *see The Sedona Conference Commentary on Legal Holds, Second Edition: The Trigger & The Process,* 20 SEDONA CONF. J. 341 (2019), *supra*; *The Sedona Conference Commentary on Proportionality in Electronic Discovery,* 18 SEDONA CONF. J. 155 141 (2017), *supra.*

[64] *Jones v. Goord*, 2002 WL 1007614 (S.D.N.Y. May 16, 2002), *claim dismissed*, *Jones v. Goord*, 435 F. Supp. 2d 221, 266 (S.D.N.Y. 2006) (denying plaintiffs' motion to compel production of database maintained by the New York State Department of Correctional Services where the state made a compelling showing that the burden of production far outweighed its benefits).

themselves (e.g., validation fields, such as those that automatically capture the user ID of the person making changes) may impede or prohibit export through standard output channels. Moreover, since many databases are intended to be used as information repositories, the system may have been designed solely with the ability for a user to add new data records or update existing records, with no functionality included for the export of records in bulk. Even extremely complex databases are often designed to be accessed by individual end users through a Graphical User Interface (GUI) through which users can view and edit a small number of records at any given time, but not the ability to export large numbers of records into a static format. To export the quantities of data often necessary to respond to civil litigation discovery requests and in a format reasonably usable to the requesting party, programmers may need to create custom tools or alternate interfaces to the database. In such conditions, the time, resources, and expense of such programming should be part of the burden analysis.

- The reporting functionality of the database: Some databases allow users to employ built-in or third-party utilities to search the database and format the results into a report that can be printed or exported as fielded data. Typically, an organization will create several standardized report templates from which the user can choose, and sometimes a system will allow users to craft custom reports. However, most reporting functions, whether template or custom, are limited in some fashion, such as in the fields that can be queried against, the number and combinations of fields that can be searched together, the volume of records that can be included in the report, or the number of characters from a given data field that can be included in the report. Additionally, certain reports may be available within a system only as screen views and not easily converted to a printable or exportable format. If a party is required to overcome these limitations in meeting their production requirements, litigation-specific reports may need to be created by programmers, requiring additional time (to create the custom reports) and resources, potentially including hard costs. Even with custom programming, it is possible that some database fields, such as system and validation fields, may not be capable of being included in a report writing function.[65]

- The extent to which a database system is in the custody of a third party: In situations where a responding party has outsourced its databases systems containing responsive ESI to offsite storage solutions under the custody of a third-party referred to as "infrastructure as a service" (IAAS), or is using a third-party software hosting repository referred to as a "software as a service" (SAAS) system (e.g., Salesforce.com), the responding party may not have the direct access to the back end of the database that is required to implement custom

[65] The reverse problem occurs when data from a legacy system or from a time before the implementation of preservation efforts exist solely in "report" format and not in the original database structure format. It may be unduly burdensome for the producing party to restore that data to the original format. Indeed, if the data is maintained only in report format in the ordinary course of business, there may be no obligation at all to convert the data into an alternate format.

programming. The parties should consider the feasibility, burden and cost of timely exporting responsive database information, and whether there is a less burdensome alternative.

- The active or legacy status of the database: Unlike unstructured data, where the trend generally is to consider active information reasonably accessible,[66] the fact that a database is in active use does not automatically mean that the data is easy and inexpensive to produce in litigation. Whether a database is active or in legacy status does not determine its accessibility. The same challenges in producing data from a database currently in use as in one that is no longer active (e.g., limited export functionality, poor data consistency, a limited-feature search engine), legacy databases can often pose additional challenges. For example, the software platform or operating system necessary to run a legacy database may no longer exist or can no longer be run on current hardware. Similarly, IT or business personnel who were familiar with the structure of the database may have left the organization, and it may be difficult, if not impossible, to find resources to export data or write any custom reports.

- The responding party's willingness to make the entire database available: In some cases, a requesting party will seek access to the entirety of a database, so that it can conduct its own investigation and generate its own reports. Absent some showing of improper discovery conduct, a responding party will generally not be ordered to make the entire database available for inspection.[67] Responding parties typically resist such requests because databases often contain large volumes of non-responsive information that may be privileged, confidential, or sensitive. However, in rare cases, a responding party may be willing to make an entire database available for inspection. In such cases, if a requesting party continues to seek new reports generated from the database, the responding party's willingness to make the entire database available should factor into the proportionality analysis.

    - The availability of database system source material, if relevant: Much of the information describing database structures and supporting hardware and software systems can be found in the end-user manuals, system documentation, written system backup procedures, training materials, and other documentation that accrues during the development or deployment of the system.

    - Legacy Systems: Finding documentation for legacy systems may prove much more difficult, as supporting materials (and knowledgeable employees) for systems not in active use are often no longer available after a period. In situations where requested supporting information for legacy systems is not available, a responding party should not be required to either create new comprehensive documentation or

[66] See Zubulake v. UBS Warburg LLC ("Zubulake I") 217 F.R.D. 309, 321-22 (S.D.N.Y. 2003); *The Sedona Conference Commentary on Preservation, Management and Identification of Sources of Information that are Not Reasonably Accessible,* (2009), supra.

[67] *See*, *e.g.*, *In re Ford Motor Co.*, 345 F.3d 1315 (11th Cir. 2003).

deconstruct the database system for the purpose of assisting the requesting party's understanding of the system and the responsive database information.

- Proprietary Systems: It also may be difficult to find comprehensive documentation for highly integrated proprietary systems, such as financial systems from SAP or Oracle, and this information may not be readily available from either the responding party or the solutions provider. Additionally, the responding party may lack actual access to certain data tables that may be a trade secret of the solutions provider, and it thus may not be possible for it to respond fully to a request for database table structure and overall organization. While the responding party should take reasonable steps to locate and produce any such relevant, but proprietary, database system information, including obtaining information from alternate sources, the courts should consider the proportionality of the burden and costs associated with licensing or otherwise locating the requested information that is not within the party's custody and control.

3. Use of Test Queries and Pilot Projects

Parties should use objective information, such as that generated from test queries, pilot projects, and interviews with persons with relevant knowledge to ascertain the burden and benefits to collect and produce information stored in databases and to reach consensus on the scope of discovery.

Comment 3:

Many disputes about the discovery of potentially relevant information stored in databases are based on deduction and inference, rather than empirical data. A requesting party may insist that certain types of information must have been stored in an opponent's database because that's what should be there. Conversely, a responding party may estimate the burden of responding to discovery requests without ever testing whether its assumptions are accurate. Neither of these approaches is acceptable. Database designs are rapidly evolving and store data in new and unique ways. Basing requests on historical database designs may cause requesting parties to miss potentially responsive data. For example, NoSQL databases do not store information in tabular form, i.e., rows and columns, so standard SQL queries will not work on such a database.

> **Illustration i**: A requesting party seeks data from a customer database. Their request asked for fields such as contact's name, email address, phone number, job title, linked organizations. They also ask for transactional data associated with the customers. However, because the requesting party used a pre-conceived assumption of what type of information a customer database would contain, they failed to request information commonly found in newer customer-oriented databases, such as behavioral data (information that builds a picture of a customer's behaviors, such as free trial signups, user account logins, feature utilization, user license additions, deactivations, and downgrades), and attitudinal data (information that represents what a customer thinks about the company and products, such as online reviews, support ticket comments, and satisfaction surveys).

A better approach for the requesting party is to first become familiar with the database architecture before submitting more specific requests for data. Database schematics, field mappings, user manuals, sample records, and test queries can help requesting parties understand the scope of information contained within the database.

The responding party can better establish the benefits and burdens of producing information stored in databases by examining objective information about the system. To this end, a responding party may want to examine documentation associated with the database and existing database table schematics. More incisively, the responding party may use the database management system to automatically generate new database table schematics, review the database change logs, guidelines or procedures for entering data, and may also execute one or more queries to test how long it takes the system to return results, the effect of those queries on the system's operation, the accuracy and quality of the data, the relevance of those results to the issues in the case, and the logistics required to export this information in a format that is reasonably useful to the requesting party. Each of these objectives—the speed of the system, impact on operations, the accuracy of the query, and the data extraction—can then be fine-tuned to improve efficiency and the overall results.

Regardless of whether the responding party concludes that the information requested is accessible, it may wish to create a test query or pilot project. The responding party can then decide whether it wants to share the results with the requesting party to demonstrate the steps that are being taken to respond to a discovery request and allow both sides to assess the usefulness and relevance of the exported information before incurring the cost of full production. The test queries may identify problems with the discovery request, such as over- or under-inclusion, or the pilot project may identify issues with preparing the data for production in precisely the format requested (allowing the producing party to propose a more efficient format for production). Sharing this information provides a common factual basis upon which the parties can re-examine the discovery requests and modify them appropriately before incurring the cost of a full production.

> **Illustration ii**: A requesting party seeks all records from a database of internal memoranda and reports that include certain key words and phrases, including the term market. Test queries indicate that the request would flag more than two-thirds of the records as potentially relevant, even though the subject at issue is narrowly focused. A review of samples taken from the market query reveals that all the sample records are, in fact, not relevant in any way to the dispute. Based on this and other information, the requesting party substantially revises its list of requested key words and phrases to eliminate certain terms that appear to generate junk results. Further sampling of the revised query results, which are much smaller than before, suggests that more than half of the records retrieved are likely relevant to the dispute.

In situations involving very large databases or multiple databases, test queries or pilot tests of the production process can be based on a subset of the data repository, consistent with the approach outlined in Zubulake v. UBS Warburg LLC[68] and elsewhere. Although the Zubulake opinions do not

[68] *Zubulake,* 217 F.R.D. 309, *supra.*

concern database information, the court's approach of using small, manageable test queries to generate empirical results from which the burden and benefit of further discovery could be determined has been widely adopted in other ESI situations, including discovery of database information.

Sharing technical or logistical information and using sampling to more effectively negotiate the scope of discovery are also consistent with guidance contained in The Sedona Conference Commentary on Achieving Quality in the E-Discovery Process (2013) and The Sedona Conference Cooperation Proclamation (2008).[69]

> **Illustration iii**: A receiver was appointed to oversee the liquidation of a company. The receiver was required to turn over all financial records associated with the company to a regulatory body, except for records related to a specific third party. The financial records were stored in a very complex relational database system comprised of hundreds of individual database tables. Staff who originally designed and maintained the system were no longer available. The receiver suggested using forensic software to run keyword searches through the fields of each database table in order to identify records to exclude. Counsel representing the third party was uncomfortable with this suggestion, fearing that the results would not be comprehensive and that some of the third party's records would be turned over to the regulator. To validate that the receiver's plan would identify all records associated with the third party, the parties agreed to test the process on a sample of database tables, and then examine the records that were not identified for exclusion.

4. Validation

A responding party should use reasonable measures to validate that its collection from the database is both reasonably complete and did not inadvertently modify the ESI.

Comment 4:

Due to the volume of information and the complexities of its organization inside databases, there are no established protocols or integrity checks (e.g., MD5 hash marking) to verify and validate the completeness and accuracy of database information collected from a larger database. However, verifying that information extracted from databases is an accurate copy of the same information as it is stored in the original database should not be seen as an insurmountable task; as a matter of due diligence, basic checks exist to ensure the completeness, accuracy, quality, and integrity of the collected data.

[69] However, a responding party is not obligated to run test queries and provide sampling information to requesting parties to satisfy curiosity. For example, when a responding party reasonably believes that a database or other structured data source contains no relevant information, it should not be obligated to sample the system absent particularized and credible evidence to the contrary. *See* Principle 6: Responsibilities of Responding Parties, *supra,* at II.C. *See The Sedona Conference Commentary on Achieving Quality in E-Discovery*, 15 SEDONA CONF. J. 265 (2014), https://thesedonaconference.org/download-pub/3668; and *see The Sedona Conference Cooperation Proclamation,* 10 SEDONA CONF. J. 331 (2009 Supp.).

> **Illustration iv**: Queries were run on a customer database, and the results included attitudinal data in the form of customer support ticket comments. The producing party did not have any formal requirements or guidelines for how customer service staff should write comments. Some customer service staff composed detailed comments, while some wrote brief notes. As a result, while the data included in the results was accurate, because the quality of the customer service comments was variable, conclusions based on this information could be flawed unless the receiving party understood the quality of the data.

Extracting data from a database in response to a discovery request typically involves: (1) executing a query to identify responsive records; and (2) structuring the responsive fields into an export format acceptable for production. Running queries and structuring output files frequently can result in unintended changes to data values, such as truncating text, substituting codes for values, or other data transformations. Other typical data extraction problems include unintentionally extracting records that are not responsive (over-inclusion) or missing records that should be included (under-inclusion) in the production set. These and other data integrity issues can render the resulting dataset incomplete or inaccurate, and thus unacceptable for production.

To reduce the risk that information extracted from databases contains transcription errors, a responding party that is extracting data from a database and formatting it into a report or file for the purpose of responding to a discovery request should test the proposed dataset to confirm that it includes all expected content and complies with the target format.

Depending on the nature of the data and the methodology used to extract the data, a variety of validation procedures may be considered:

- Validating numeric values. When data consists of a numeric value, the following tests may be appropriate:
  - Confirm that the number of extracted database records matches the number of records that were originally identified by one or more target queries.
  - Compare the resulting number of records to the number that appears in reports that are regularly produced in the ordinary course of business.
  - Compare the number of extracted records to control counts from the tables being queried.
  - Compare the aggregate of certain fields, such as sales amounts, to known control totals from routine or regularly produced reports.
  - Develop control totals by confirming that the sum of the extracted records plus the total of the non-extracted records equals the total of the same field or record set as noted in the entire table or report.

Illustration v: A party requests information about all buyers of a product, including the date of purchase, the price paid, and the state in which the purchase took place. All this information is tracked in a sales database maintained by the responding party. The responding party runs a query to identify all the sales records for that specific product and exports the requested information into a .CSV file. Before producing this information, the responding party double-checks the number of data rows in the .CSV file by loading it into a spreadsheet program and comparing the number of lines to the number of records identified in the query. The responding party then checks to make sure that the date and price field rows contain only date and price information. Satisfied with the results of these checks, the responding party then provides this information to the requesting party.

- Validating formatted values: Data often consists of numeric values in formatted fields, depending on the purpose of the data. Examples of formats often used in databases are: currency, date, time, percentage, fraction, telephone numbers, and zip codes. When data consists of formatted values, the following tests may be appropriate:
    - Confirm that the formatted values are accurate and consistent with other values in the record and database.
    - Confirm that the values are accurately and consistently formatted in all extracted records.
    - Confirm that formatting is not lost in the extracted data.
    - Validating standard language values: When the contents of an extracted field consist of standard language values rather than a numeric value, the responding party should confirm that the extracted text values conform to a list of expected values for those fields. For example, for fields that can contain only a limited number of valid values, such as the seven days of the week or the twelve months of the year, a responding party can run an automated comparison of the extracted information against all possible expected values for these fields to ensure that no unexpected values are included.
- Validating non-standardized language values: When text fields do not require standardized language, as in many narrative or comment fields, a sample of fields from the extracted text can be examined to confirm that the information meets expectations of the information that should be stored there. Samples of extracted text fields can also be compared to the corresponding records in routine or regularly prepared reports to confirm that the extracted text field information is consistent with presentation of the same information in validated reports used in the ordinary course of business.
- Validating from multiple fields: In situations where values in the production dataset are calculated from several fields in the source database, responding parties can help make the

extracted fields more easily validated by including not only the field containing a calculated result field value, but also the source field values from which the resultant values are calculated. Including this additional information would make it possible for both requesting and responding parties to check the internal consistency of the final result field.

- Validating from multiple tables (relational databases): In relational databases, multiple tables of data are often linked by key values that are echoed on one or more tables. Extracted database information that has either been retrieved from or is being produced in multiple tables can be checked for accuracy and completeness by confirming that the linking key values from the various tables are consistent and sufficient to properly link the records from the various tables. Ambiguous key values—i.e., values that do not provide a unique relationship between correct data elements—can occur when information is extracted from multiple tables.

- Validating from reports: Responding parties should not underestimate the ability of database reports in general to confirm the accuracy of a data extraction. Many standard reports that are used on a regular basis within an organization, including regulatory filings generated through queries or scripted tools, compile sophisticated information and metrics that can be used to double-check the accuracy and consistency of many types of data fields extracted from a database.

- Validating from query design: Finally, requesting parties may ask for the actual queries used to generate the results. This would allow them to determine if any type of data modification (such as truncating the size of an output field) or data concatenation (merging data from more than one original field into a single output field) was carried out.

Authenticating exported database information builds on validation processes, and more than one procedure can be used to demonstrate sufficient consistency, completeness, and accuracy in the extracted data. However, situations can occur in which field values are different in the source database and in the extracted data. Typically, such differences are caused by mechanical issues, such as a report template that truncates the information in a field after the first N characters, thereby displaying only a partial entry that cannot be fully validated against the original database input. However, if these differences are not caught soon after the extracted data has been prepared and produced, the consequences of relying upon the extracted data can have far-reaching consequences. Both requesting and producing parties should consider adding quality assurance procedures to ensure that such errors are quickly identified.

5. Data Authenticity and Admissibility

The proper validation of collection from a database does not automatically make the substantive information stored in the database authentic, admissible or true. These are separate issues that need to be analyzed by the appropriate parties.

Systems or components can malfunction, errors may occur in programs and formulas, manual data entry may introduce errors, and certain cells, fields or tables can be mislabeled or misinterpreted, the way that certain fields within a database are used may change over time, meaning that old data records and new data records may use the same fields but record different information.

Most often, database information will be introduced in evidence as a business record under Fed. R. Evid. 803(6), which sets forth five conditions for a business record to be considered admissible:

1. The record was made at or near the time by—or from information transmitted by—someone with knowledge;
2. The record was kept in the course of a regularly conducted activity of a business, organization, occupation, or calling, whether or not for profit;
3. Making the record was a regular practice of that activity;
4. All these conditions are shown by the testimony of the custodian or another qualified witness, or by a certification that complies with Rule 902(11) or (12) or with a statute permitting certification; and
5. The opponent does not show that the source of information or the method or circumstances of preparation indicate a lack of trustworthiness.

While these conditions are often cited, there is a significant disparity in the approaches for admitting database information as evidence. As such, parties should be prepared to establish a solid foundation for the evidence, whether through custodian testimony or stipulation.

Even if the conditions of Fed.R.Evid. 803(6) are met, the court must also be persuaded that the records are authentic. However, "[a] person complying with a discovery request in a civil case or subpoena in a criminal case implicitly avers that the matter produced is the evidence requested."[70][71]

A key authenticity issue in business records is what has or may have happened to the record or records in the intervening time between record creation and trial. Under Fed.R.Evid. 901(a), one must demonstrate that the record that has been retrieved from the file is the same as the record that was

[70] *5 Weinstein's Federal Evidence* § 900.07 (2021); see also *United States v. Brown*, 688 F.2d 1112, 1116 (7th Cir. 1982).

[71] The 2014 edition of the Database Principles cited *In re Vinhnee*, 336 B.R. 437, 446-47 (B.A.P. 9th Cir. 2005). However, the courts generally have been more comfortable with authenticating ESI over the past decade. *Sedona Conference Database Principles*, 15 SEDONA CONF. J. 171, 212-214 (2014); *see also Sedona Conference Commentary of ESI Evidence & Admissibility*, Second Edition, 22 SEDONA CONF. J. 83, 96-97 (2021).

originally placed into the file. Database records by nature and design are updated regularly. Records created at or near the time relevant to the litigation may not be the same at the time of trial.[72]

Parties may want to consider using the Rule 26(f) conference to identify and discuss such systems and integrate into the proposed discovery plan to reduce future discovery costs and trial time. Admissibility and authenticity, however, do not equate to accurate and true. While databases are often relied upon in the ordinary course of business, these systems are no more reliable than other types of data. Many factors can impact the accuracy of the underlying data. Current enterprise systems may be comprised of multiple, complex, integrated systems. Identifying the appropriate system of record is essential to avoid unnecessary requests from transient, intermediary, or aggregated data systems.

These enterprise systems or components can malfunction, errors may occur in programs and formulas, manual data entry may introduce errors, and certain cells, fields or tables can be mislabeled or misinterpreted. Most users interact with databases through an application layer. This application layer may aggregate, convert, or calculate values from the database to present the information in a more digestible fashion. In addition, the way that certain fields within a database are used may change over time, meaning that old data records and new data records may use the same fields but record different information. While rare, it is also possible that data may have been intentionally or unintentionally manipulated (through data conversion, aggregation calculation or other) in a way that degrades the quality of the data being produced. Such degradation may take the form of data that lacks certain metadata fields that are integral to understanding the remainder of the information.

Understanding these systems and discussing them during the Rule 26(f) conference can guide the development of the discovery plan, including data integrity and collection methodologies, to avoid costly downstream data issues.

6. Form of Production

The way in which a requesting party intends to use database information is an important factor in determining an appropriate format of production.

B. Sedona Principle 3: Confer and Seek Agreement

The Federal Rules and associated committee notes, consistent with Sedona Conference's Principles, provide guidance that should avoid serious contests over the form of production of database content in discovery. This starts with the required discovery conference under Rule 26(f) where the parties must attempt "in good faith to agree on the proposed discovery plan" and discuss the discovery of ESI "including the form or forms in which it should be produced." Fed.R.Civ.P. 26(f)(2) and (3)(c).

[72] Database tables often have fields that identify when a record was created, when modified, and by whom. Parties may want to include such fields in their request for production.

The committee notes further explain:

> When a case involves discovery of electronically stored information, the issues to be addressed during the Rule 26(f) conference depend on the nature and extent of the contemplated discovery and of the parties' information systems. It may be important for the parties to discuss those systems, and accordingly important for counsel to become familiar with those systems before the conference. With that information, the parties can develop a discovery plan that takes into account the capabilities of their computer systems. In appropriate cases, identification of, and early discovery from, individuals with special knowledge of a party's computer systems may be helpful.
>
> Early discussion of the forms of production may facilitate the application of Rule 34(b) by allowing the parties to determine what forms of production will meet both parties' needs. Early identification of disputes over the forms of production may help avoid the expense and delay of searches or productions using inappropriate forms.

Committee Notes on Rules—2006 Amendment

Use of ERDs and Data Dictionaries where available would be extremely helpful for discussing production of database data. Familiarizing yourself with these or having them ahead of the conference would help guide discussions. Parties should not only become familiar with the systems but the overall workflow of how records are created, inserted, updated, and managed. Discussion of the form of production should include the export file type (Excel, CSV, TXT, etc.), defined delimiters, and text qualifiers. Parties should consider a sample production of data so that the production form can be tested and confirmed before producing all the data. Involving technical representatives from each side can often avoid litigation gamesmanship and focus the discussion on practical solutions.

Thereafter, requests for production of database information are governed by Rule 34 where the requesting party "may specify the form or forms in which electronically stored information is to be produced."[73] The responding party may object to the requested form but must state the form it intends to use.[74] As the committee notes suggest, "[s]tating the intended form before the production occurs may permit the parties to identify and seek to resolve disputes before the expense and work of the production occurs." If the requesting party does not specify a desired form, it must be produced "in a form or forms in which it is ordinarily maintained or in a reasonably usable form or forms."[75]

[73] FED. R. CIV. P. 34(b)(1)(C).

[74] FED. R. CIV. P. 34(b)(2)(D).

[75] FED. R. CIV. P. 34(2)(E)(ii).

C. Sedona Principle 12: Produce as Ordinarily Maintained or in a Reasonably Usable Form

Generally, "as ordinarily maintained" is synonymous with production in native format. However, as discussed supra in Section II.D.1, production of database information in native format is frequently not desirable for either the requesting or responding party. This is especially true for large, enterprise databases in proprietary installations such as Oracle or SQL Server. The scope of relevant data is almost always less than the whole database and setting up a database instance can be difficult.[76] Moreover, significant software and hardware costs can be involved. As a result, database information is most often produced by exporting relevant data to a "reasonably usable form."

A common, 'reasonably usable' form for production of database information is known as "fielded tables" where "where each row represents a database record and each column represents a single data field." See supra, Section II.D.3. This form is also known as near native. Information in such a format is frequently produced in Excel or in delimited text files.

As one commentator argued, the default form of production for database information should be "a fielded and electronically searchable format preserving metadata values, keys and field relationships."[77] Ball explains:

Near-native forms are sufficiently similar in content and structure to permit the near-native forms to be brought back into the native application with little or no loss of content or utility. It's possible because the structure of the data (a.k.a the fielding of the data) can be easily mapped back and forth between the native and near-native iterations.

Exports of information from database applications are often produced in so-called "delimited" formats not native to the database, but which nevertheless support the ability to interpret the exported data in ways faithful to the native source.[78]

Perhaps the main reason the fielded tables is the recommended form of production for database information is that databases are structured. The fields in a database table are structured in both form and content. The form is limited by allowing only data of a particular type, such as character,

[76] When the requesting party is technically proficient and multiple related tables have been requested, the parties can consider setting up a database, possibly leveraging available technologies such as cloud services. If a party has the onsite infrastructure, such as MS SQL Server or Oracle, creating a shell database with the ability to automatically expand and adjust as data is imported is a relatively simple exercise. If the infrastructure does not exist, cloud services such as Azure and AWS, can set up a secure database server to your specifications for a monthly fee.

[77] Craig Ball, *A Guide to Forms of Production* at 32, BALL IN YOUR COURT, (May 12, 2014), https://craigball.net/2014/05/19/a-guide-to-forms-of-production/ (last visited July 7, 2025).

[78] *Id.* at 6-7.

numeric, or date values;[79] the content is limited by the field's function in the database schema. The tables are structured by having key fields that define the relationship between tables. This contrasts with an MS Word document where the user can enter any kind of data anywhere in the file.[80]

The structure is a critical part of the information requested. Failing to preserve that structure degrades the information, something the rules committee warned against:

> [T]he option to produce in a reasonably usable form does not mean that a responding party is free to convert electronically stored information from the form in which it is ordinarily maintained to a different form that makes it more difficult or burdensome for the requesting party to use the information efficiently in the litigation. If the responding party ordinarily maintains the information it is producing in a way that makes it searchable by electronic means, the information should not be produced in a form that removes or significantly degrades this feature.

Committee Notes on Rules—2006 Amendment.[81]

As the court explained in Local 3621 EMS Officers Union DC-37 AFSCME AFL-CIO v City of New York, 2024 WL 1856302, 2024 U.S. Dist. LEXIS 79990 (S.D.N.Y. 2024):

The Sedona Principles on Databases explicitly contemplate that when a large portion of a data set is not responsive, custom queries may need to be used to obtain the data.[82]

"[A] producing party is required to 'provide the requesting party a functionally adequate ability to access, cull, analyze, search, and display the ESI, as may be appropriate given its nature and the proportional needs of the case.' (emphasis added).[83]

[79] When production is made in CSV or TXT format, care should be taken on import to maintain the database's field type so that, for example, numbers or dates are imported correctly.

[80] Spreadsheets are examples of semi-structured files: a user can maintain a structure and, for example, enter only valid dates in a date column, but the software does not enforce that limit.

[81] In *Jannx Med.Sys., Inc. v. Methodist Hosps.*, 2010 U.S. Dist. LEXIS 122574 (N.D.Ind.11-17-2010), Plaintiffs produced database data in .pdf format. Defendants filed a motion to compel stating that the production of this data in .pdf format was in violation of Rule 34. Plaintiffs asserted that producing documents in .pdf format was in compliance with Rule 34 because Defendants did not specify the exact form in which the documents were to be produced. The Plaintiff did not argue that the production of electronic database data in .pdf form maintained the Defendants ability to search the information. The Court granted the Defendants motion to compel to the extent that Defendants request that Plaintiff produce responsive information in an electronic database format that allows the information to be reasonably usable, i.e., fully searchable and manipulable, with the connections between the data fields intact.

[82] *See* The Sedona Conference*, The Sedona Conference Database Principles Addressing the Preservation and Production of Databases*, 15 SEDONA CONF. J. 171, 181 (2014).

[83] *Local 3621 EMS Officers Union*, 2024 WL 1856302, at *3-4, 2024 U.S. Dist. LEXIS 79990, at *7.

IV. APPENDIX: MOST COMMON DATABASE PLATFORMS

Introduction

There are many types of databases in use today. A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system ("DBMS"). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database. Database management has evolved significantly, and this appendix will cover the most common database platforms used in business today. To illustrate how these platforms work, this appendix uses examples of databases containing data about cars.

Relational Database

Relational Database Management Systems ("RDBMS") are the most common and rigid of all database types. RBDMS was created for mapping datasets at IBM in 1970. RDBMS is the basis for common database platforms such as SQL Server, Oracle, and MySQL.

Data in an RDBMS is organized in tables, columns, and rows. Each table contains data that is stored in rows and columns. Each row is considered a single record listed horizontally. Each column is a field that describes what is being stored which is stored vertically. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

Tables and rows in a relational model are referenced via keys—primary and foreign. A primary key uniquely identifies a row. Foreign keys link to a primary key of another table, allowing two tables to interact based on linking the primary key of one to the foreign keys in another.

RDBMS make up more than 70% of all databases in production as of 2022, though that number is dropping due to the emergence of other database types that help meet the needs of dynamic data storage and processing as the volume, variety, and velocity of data in the world constantly increase. Relational databases are used for a wide variety of applications and are frequently the underlying technology in legacy systems.

Below are example tables from a database containing information about car brands:

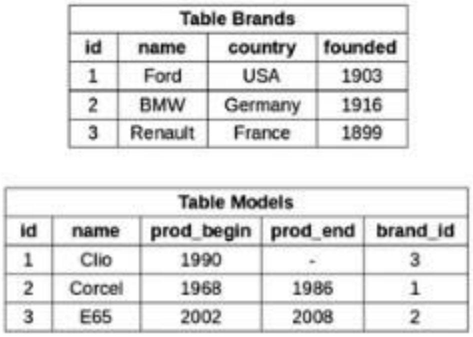


Table Brands			
id	name	country	founded
1	Ford	USA	1903
2	BMW	Germany	1916
3	Renault	France	1899

Table Models				
id	name	prod_begin	prod_end	brand_id
1	Clio	1990	-	3
2	Corcel	1968	1986	1
3	E65	2002	2008	2

Image source: https://www.researchgate.net/figure/Cars-RDB-schema-a-and-the-corresponding-schema-in-the-canonical-model-b_fig1_334118633

The following represents how tables in a database containing car information can relate to one another:

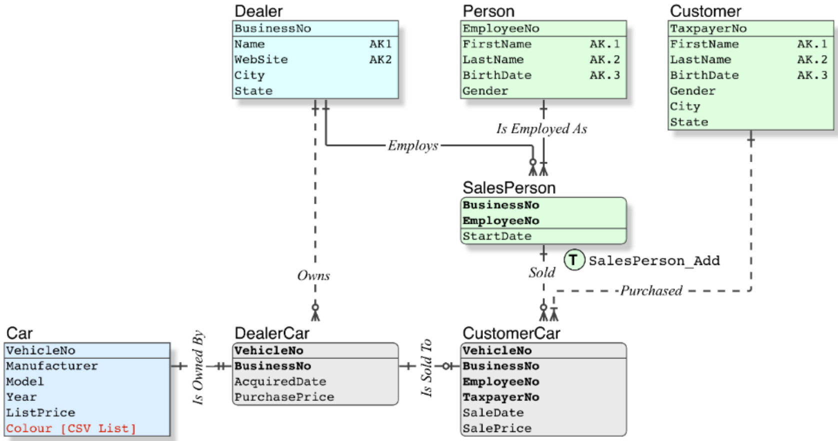


Image source: https://stackoverflow.com/questions/59132125/er-relational-table-to-database

Object-Oriented Database

Object-Oriented Database Management Systems (OODBMS), as the name suggests, store data in objects. OODBMS are often used in devices, compiled software, and real-time systems.

OODBMS are different in structure from relational databases. Because the world rarely looks like an RDBMS' tables, objects can be designed and implemented to reflect reality more closely.

The contrast between an RDBMS and an OODBMS can be illustrated by returning to the example of a database containing information about cars. In a RDBMS, car data is divided across multiple tables, like parts details, maintenance records, and manufacturers, each identified with unique numbers. This structure simplifies data updates as changes in a single place suffice.

On the other hand, an OODBMS treats a car as a class, storing all related information like parts and ownership history in a single instance of that class. This related data stored within the class object are referred to as attributes. This class outlines the data storage for its objects, and each object can utilize some or all features of the class.

Object-oriented databases are devised to hold objects from object-oriented programming environments, contrasting with table storage in relational databases. Here, an object is identified by its class, and operations are carried out via methods as opposed to queries, with data grouped by object—an approach that can impact both data collection and production. Where an RDBMS is designed in advance with a specific schema, an object-oriented database may allow for user generated schema. Object-oriented databases use Object Query Language (OQL) rather than SQL. Unlike SQL, which works with tables and rows, OQL works with data as objects, making it easier to find and manage specific pieces of data within complex structures.

Object-oriented programming languages are designed for speed as they allow for the reuse of code and easy maintenance, which accelerates development and execution processes. In object-oriented programming, every object has a unique ID, which is like its personal address that can be pointed to directly, making it faster and easier to find specific pieces of data when needed. OODBMS are especially useful for managing large binary objects like images and videos because they allow these complex data types to be stored, retrieved, and manipulated more efficiently as single objects, rather than splitting them across multiple tables.

Example OODBMS engines include ObjectDB, Objectivity/DB, and Versant.

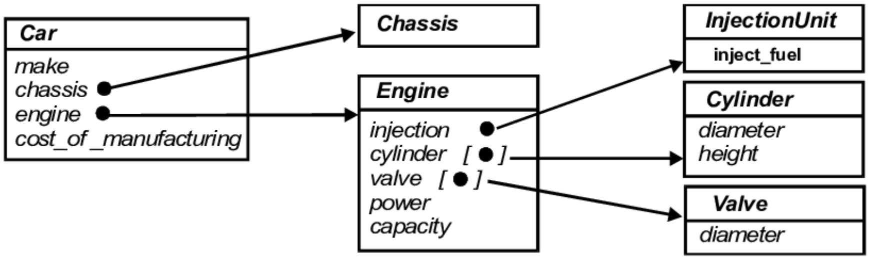


Image source: https://www.researchgate.net/figure/View-schema-integrating-information-about-designs-of-engines_fig2_2517827

NOSQL Databases

Since database storage is a technique for organizing, storing, and accessing data, different types of data may require different types of databases. NoSQL databases, including key-value stores, document stores, column-oriented databases, and graph databases, store their data differently than an RDBMS does. In general, a NoSQL database is any database that breaks away from the traditional design of SQL. Examples include Apache HBase, Google's Bigtable, and Scylla.

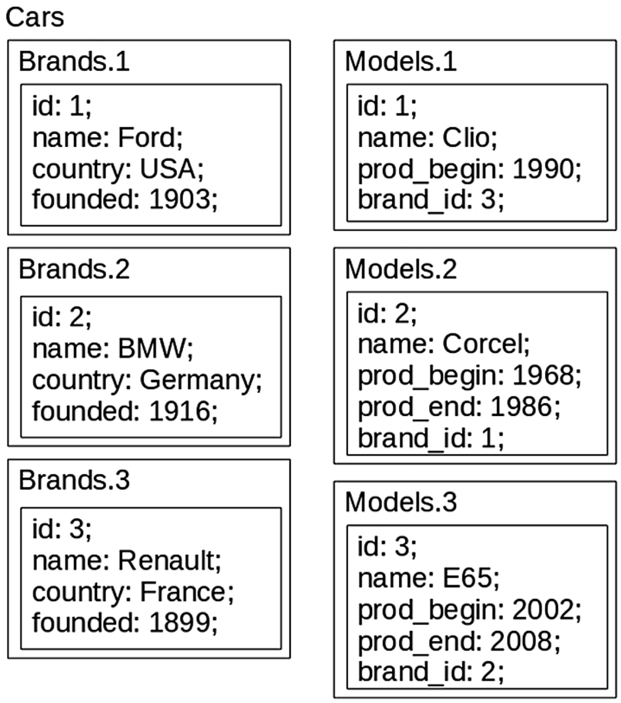
Key-Value Database

A key-value database, the most basic type of NoSQL database, functions on two elements: a unique key and its associated value. Unlike in an RDBMS, the values here are not type-restricted and cannot be commonly queried with SQL.

While it lacks a traditional schema, a key-value database organizes keys into a keyspace for easy retrieval through exact key matching. Every record in this database has a unique key, and every object is linked to one of these keys. A value in a key-value database can be anything so long as it is tied to a unique key—a significant deviation from an RDBMS' schema.

The simplicity of a key-value database makes it difficult to search on a per-field basis because storage is explicitly and exclusively tied to the single key for each record. Key-value databases can be utilized over multiple servers, in both partitioned and mirrored environments, but may pose evidence-related issues if the database's state is relevant. A key-value database is easy to update and change due to the simplicity of the structure of the system. However, a value may only be retrieved by a specific key and depending on the way the key-value database is deployed, the value may not be searchable at all.

Examples of key-value database engines include Apache Cassanda, Redis, and Amazon DynamoDB.



Cars

Brands.1
id: 1;
name: Ford;
country: USA;
founded: 1903;

Brands.2
id: 2;
name: BMW;
country: Germany;
founded: 1916;

Brands.3
id: 3;
name: Renault;
country: France;
founded: 1899;

Models.1
id: 1;
name: Clio;
prod_begin: 1990;
brand_id: 3;

Models.2
id: 2;
name: Corcel;
prod_begin: 1968;
prod_end: 1986;
brand_id: 1;

Models.3
id: 3;
name: E65;
prod_begin: 2002;
prod_end: 2008;
brand_id: 2;

Image source: https://www.researchgate.net/figure/A-key-value-schema-generated-by-the-mapping-of-the-canonical-schema-from-Fig1b_fig4_334118633

Document Database

One common variety of key-value databases is a document database. A document database is a key-value database in which the keys are unique values, and the values are documents which are comprised of structured or semi-structured data. Each document is a single database record.

Documents in an NoSQL Document Database store data in key-value pairs within the document itself, in addition to being the "value" in the key-value pair which defines the database.

Data in a document database can be retrieved by ID (the key) and data can be indexed by fields within a document.

One important feature of a document database is its flexible schema. One significant differentiating feature between a standard key-value database and a document database is that the fields within a document database are indexed and thus can be queried, allowing for field-based evaluation and querying, where standard key-value databases do not. This allows for the retrieval of data subsets within a document.

With a standard key-value database, the result of a search is the entire value which cannot be searched until it is exported from the database. An important distinction between document and relational databases is that all data for each document is the entirety of data about that document—there is no mechanism for joining data points together as one can in a relational database.

Document databases often store their data in JSON and are often used for Content Management Systems, eCommerce sites, and analytics.

Example systems include MongoDB, CouchDB, and Amazon's DocumentDB.

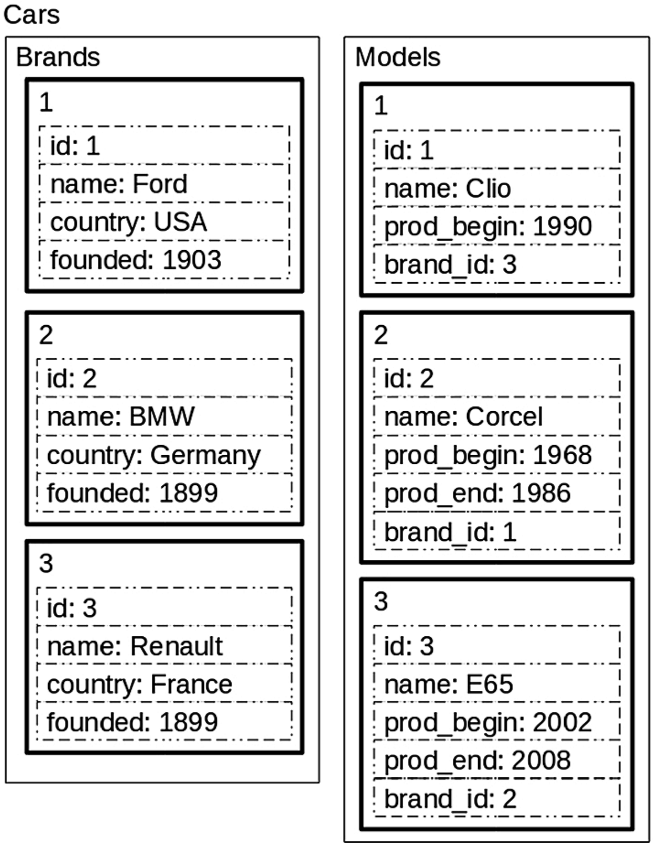


Image source: https://www.researchgate.net/figure/A-document-oriented-schema-generated-by-the-mapping-of-the-canonical-schema-of-Fig1b_fig5_334118633

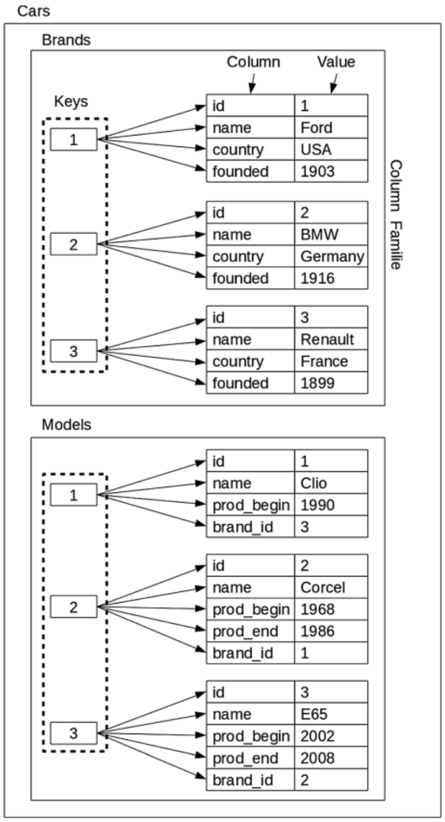
Columnar Database

A column store database, or a columnar database, is a type of database that stores data by columns instead of by rows as is commonly done in traditional databases. This means that all data for a particular field is stored as a single unit—a column—enabling faster and more efficient vertical search and retrieval of data in some applications. This is especially useful in large data sets.

Columnar databases are frequently used for analytical queries and data warehousing where calculations are performed over a single field of data. In addition to storing data vertically in a column, data is also queried and fetched column-wise, thereby reducing the time in finding specific data in a single field. This database type works well when quick data analysis results are needed from a huge dataset.

As a practice pointer, where a traditional row-oriented database record is comprised of multiple fields and document requests focus on the schema of those records, in a columnar database all entries for a field are stored together and independent of the other fields within the database. This organization method makes it easy to query, compile, and analyze all information related to a specific field without the overhead of the other fields in a record.

By way of example, in a row-oriented database storing information about cars, the manufacturer, year, and owner would be stored in a single record. To analyze all the manufacturers in a row-oriented database, a significant amount of data needs to be ignored from each row, which is both inefficient and time consuming. A columnar database, by contrast, stores all the manufacturers of the specific vehicles in a column together, allowing for easy analysis of that single feature without reference to anything else.



Column	Value
id	1
name	Ford
country	USA
founded	1903

Column	Value
id	2
name	BMW
country	Germany
founded	1916

Column	Value
id	3
name	Renault
country	France
founded	1899

id	1
name	Clio
prod_begin	1990
brand_id	3

id	2
name	Corcel
prod_begin	1968
prod_end	1986
brand_id	1

id	3
name	E65
prod_begin	2002
prod_end	2008
brand_id	2

Image source: https://www.researchgate.net/figure/A-column-oriented-schema-generated-by-the-mapping-of-the-canonical-schema-of-Fig1b_fig6_334118633

Wide-Column Store

Building on the columnar database concept, a wide-column store, also known as a column-family database, is a type of NoSQL database that stores multiple columns together in a structure called a column family. Unlike traditional databases, it allows storing of various data types in different columns, and the columns can vary from row to row within the same column family, adding flexibility.

This type of database is great for analyzing large datasets and is designed to handle heavy write loads. It is often used in systems where high performance, scalability, and flexibility are crucial because it has the capacity to hold billions of rows and millions of columns.

Extending the previous vehicle example, a wide-column store might store columns for the make, model, and color of a vehicle, rather than all relevant information as would be stored in a row-oriented database or just the make or model in a columnar database. This enables easier analysis of the grouped columns without the overhead of a row-oriented database but with the complexity of multiple columns.

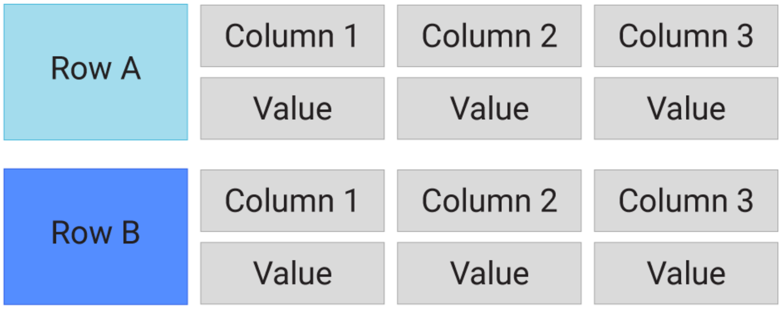
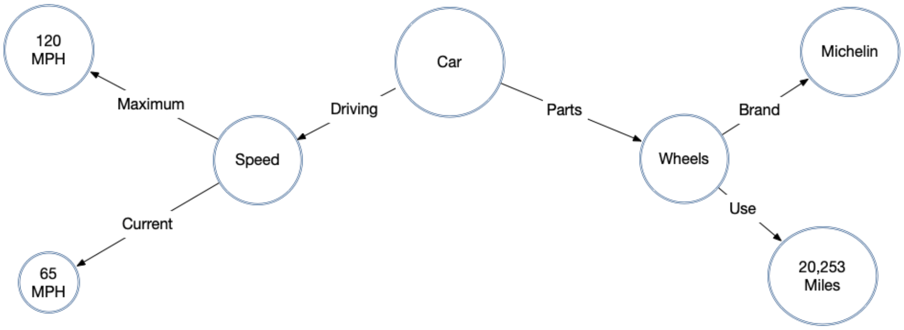


Image source: https://www.scylladb.com/glossary/wide-column-database/

Graph Database

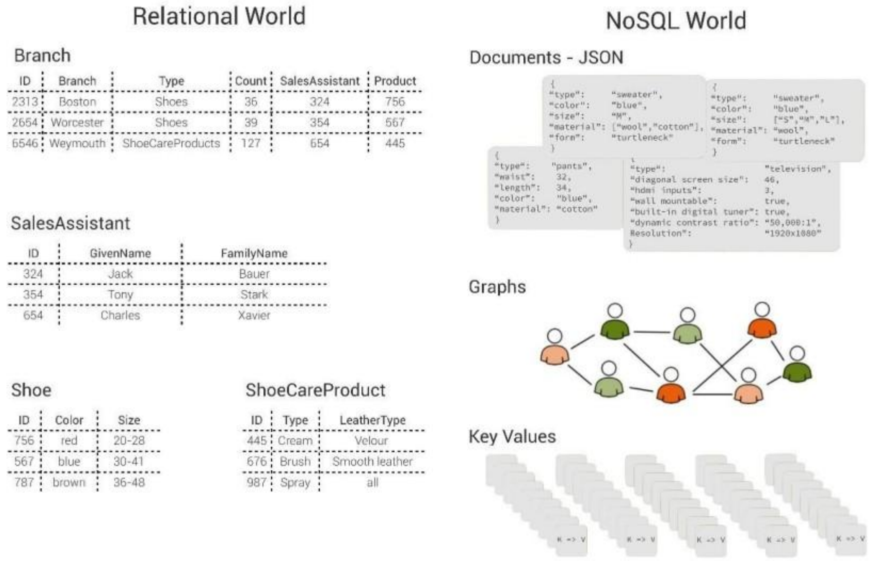
A graph database is a type of NoSQL database which has an entirely different metaphor than an RDBMS or any other database type described supra. Graph databases do not use columns or rows and therefore have no schema. Rather, each entry is a set of discrete objects called a node which is related to zero or more of the other objects within the database via edges. When visualizing the structure, a molecular diagram may be of use. Each node contains some data, called a property.

A graph database is often used to measure and describe relationships between things. The real world is rarely as well organized as a database table's columns and rows, and a graph database is designed to focus on the interconnectedness of things—whether objects, people, or data. Graph databases are used to explore relationships and are used extensively for fraud detection, social network investigations, and to drive recommendation engines. Although the most important feature of a graph database is the relationship between nodes, it should not be confused with a relational database.



Multi Model Database

A multi-model database is a data management system that supports multiple data models, which define the parameters for how data can be organized and arranged. This might include document, graph, relational, and key-value formats. Integrating multiple models in a single database allows handling of various data types and can reduce the need to deploy multiple databases. These databases thus offer great flexibility and are equipped to meet diverse application requirements efficiently. They are typically used in applications where high levels of diversity and flexibility in data handling are required.



Relational World

Branch

ID	Branch	Type	Count	SalesAssistant	Product
2313	Boston	Shoes	36	324	756
2654	Worcester	Shoes	39	354	567
6546	Weymouth	ShoeCareProducts	127	654	445

SalesAssistant

ID	GivenName	FamilyName
324	Jack	Bauer
354	Tony	Stark
654	Charles	Xavier

Shoe

ID	Color	Size
756	red	20-28
567	blue	30-41
787	brown	36-48

ShoeCareProduct

ID	Type	LeatherType
445	Cream	Velour
676	Brush	Smooth leather
987	Spray	all

NoSQL World

Documents - JSON

```
{
"type":     "sweater",
"color":    "blue",
"size":     "M",
"material": ["wool","cotton"],
"form":     "turtleneck"
}
```

```
{
"type":     "sweater",
"color":    "blue",
"size":     ["S","M","L"],
"material": "wool",
"form":     "turtleneck"
}
```

```
{
"type":     "pants",
"waist":     32,
"length":    34,
"color":     "blue",
"material": "cotton"
}
```

```
{
"type":                       "television",
"diagonal screen size":       46,
"hdmi inputs":                3,
"wall mountable":             true,
"built-in digital tuner":     true,
"dynamic contrast ratio":     "50,000:1",
Resolution":                  "1920x1080"
}
```

Graphs

Key Values

Image source: https://arangodb.com/wp-content/uploads/2020/03/ArangoDB-White-Paper_What-is-a-multi-model-database-and-why-use-it.pdf?hsCtaTracking=964a2732-53d1-477e-93ed-0e7430c8d1bf%7C7ff1d46f-2bc6-439e-8e69-98b650993860

Data Exchange Formats

Transferring data between databases and production in litigation is often done via data exchange format files. Data exchange format files are platform independent text files which contain content, often extracted from a database or other structured data source. These files are designed to be software agnostic, standard file formats, and are well suited for export and production from databases.

JSON

JavaScript Object Notation (JSON)[84] is an open-source data exchange format used for data transfer and storage. JSON's data is organized in object-oriented key-value pairs. It's based on the format of JavaScript object literals and used to store simple data structures and objects in a standardized and text-readable format, often used in web applications for data transfer.

XML

An Extensible Markup Language (XML)[85] file is a universally accepted standard format for storing and transferring data. It is characterized by its use of custom tags to define the structure and nature of the contained data, making it easy to understand and widely used for data exchange between applications.

CSV

A comma-separated value (CSV)[86] file is a text file used to transmit fielded data. The fields are indicated or delimited by an identified character, often a comma, a pipe ("|"), or other unique character.

Other Text File Formats

Tab-delimited and fixed-width files are data files where each line represents a new record and fields within a record are separated by tabs, or aligned to fixed positions, allowing for precise organization and easy readability of data. These types of files are similar to CSVs and provide an uncomplicated format for transmitting data between programs.

[84] Douglas Crockford, *Introducing JSON*, JSON https://www.json.org/json-en.html (last visited June 17, 2025).

[85] *Extensible Markup Language (XML), W3C*, WORLD WIDE WEB CONSORTIUM, https://www.w3.org/XML/(last visited June 17, 2025).

[86] Yakov Shafranovich, *Common Format and MIME Type for Comma-Separated Values (CSV) Files*, RFC 4180 (Oct. 2005), https://datatracker.ietf.org/doc/html/rfc4180.

Other Data Exchange Formats

Excel®

An Excel file is a spreadsheet document often used for storing, organizing, and manipulating numerical and text data within a grid of cells. Excel files are also used to transmit fielded data, like a CSV. Unlike CSVs, Excel files can only be fully utilized with specific software while CSV files can be opened by any text editor or database engine. Excel should be used with caution as a database exchange format as most spreadsheet engines add, modify, and delete metadata from Excel files.