

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ACTIFIO, INC.,
Petitioner,

v.

DELPHIX CORP.,
Patent Owner.

Case IPR2015-00128¹
Patent 8,468,174 B1

Before JENNIFER S. BISK, PATRICK R. SCANLON, and
MINN CHUNG, *Administrative Patent Judges*.

BISK, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

¹ Case IPR2015-00136 has been consolidated with this proceeding.

INTRODUCTION

A. Background

Petitioner, Actifio, Inc., filed two Petitions requesting *inter partes* review of claims 1, 4, 5, 27–29, and 34 (the “challenged claims”) of U.S. Patent No. 8,468,174 B1 (Ex. 1001, “the ’174 patent”). IPR2015-00128, Paper 1 (“Pet.”); IPR2015-00136, Paper 1 (“’136 Pet.”).² On April 29, 2015, we consolidated the two proceedings and instituted an *inter partes* review based on the following grounds of unpatentability (Paper 7, “Inst. Dec.”):

References	Challenged Claims
Edwards, ³ Edwards II, ⁴ Neto, ⁵ and Klivansky ⁶	1, 27, and 28
Edwards, Edwards II, Neto, Klivansky, and Hart ⁷	4, 5, 29, and 34

This is a Final Written Decision under 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. For the reasons set forth below, Petitioner has shown by a

² Because of the substantial overlap in the two Petitions, we will cite only to the Petition of IPR2015-00128 unless otherwise noted.

³ John K. Edwards et al., *FlexVol: Flexible, Efficient File Volume Virtualization in WAFL*, 2008 PROC. OF THE 2008 USENIX ANNUAL TECHNICAL CONF. 129. Ex. 1003 (“Edwards”).

⁴ U.S. Patent No. 7,409,511 B2 issued Aug. 5, 2008. Ex. 1005 (“Edwards II”).

⁵ ANAND RANGANTHAN & ANTONIO JOSE RODRIGUES NETO, TECHNICAL REPORT, SNAPMANAGER 3.0 FOR ORACLE BEST PRACTICES, TR-3761 (2009). Ex. 1004 (“Neto”).

⁶ MIROSLAV KLIVANSKY, TECHNICAL WHITE PAPER, A THOROUGH INTRODUCTION TO FLEXCLONE™ VOLUMES, TR3347 (2004). Ex. 1006 (“Klivansky”).

⁷ U.S. Patent Application Pub. No. 2008/0307345 A1 published Dec. 11, 2008. Ex. 1007 (“Hart”).

preponderance of the evidence that claims 1, 4, 5, 27–29, and 34 are unpatentable.

B. Related Matters

The parties indicate that the '174 patent is involved in the lawsuit *Delphix Corp. v. Actifio, Inc.*, Case No. 5:13-cv-04613-BLF (N.D. Cal.). Pet. 2; Paper 5⁸, 1.

C. The '174 Patent

The '174 patent describes computer systems and methods for efficiently managing multiple copies of databases. Ex. 1001, 1:14–16. Specifically, the '174 patent recognizes the growing importance of databases to an organization's information technology infrastructure and the increasing amount of resources required to manage these databases. *Id.* at 1:16–31. A typical information technology infrastructure includes production database servers that run applications managing the daily transactions of the organization. *Id.* at 1:32–34. Routine changes and upgrades to a production database or the applications that work with that database typically require the use of copies of the production databases in order to protect the production environment. *Id.* at 1:34–37. Depending on the process used for making the change or upgrade at issue, this may involve several copies of the production database—one for each stage of development such as testing, certification, and training. *Id.* at 1:37–44. Because production databases are typically large and complex, the practice of making multiple full copies of these databases is expensive and inefficient. *Id.* at 1:44–52.

⁸ Because of the substantial overlap in the two proceedings, unless otherwise noted, all citations of Papers and Exhibits are from IPR2015-00128.

The '174 patent recognizes these problems and proposes creating “virtual databases” that share information so that multiple copies of database information are made only if necessary. *Id.* at 2:67–3:32. “Systems and methods for creating and using virtual databases are disclosed in [U.S. Patent No. 8,150,808 (“the '808 patent”)], which is incorporated by reference in its entirety.” *Id.* at 4:6–9.

The '808 patent describes one virtual database embodiment represented by Figure 2a, “production database system 110 . . . is the source of the database being virtualized” to create virtual database 220 using virtual database files stored in database storage system 100. *Ex.* 2002, 6:59–65. Figure 2a of the '808 patent is reproduced below.

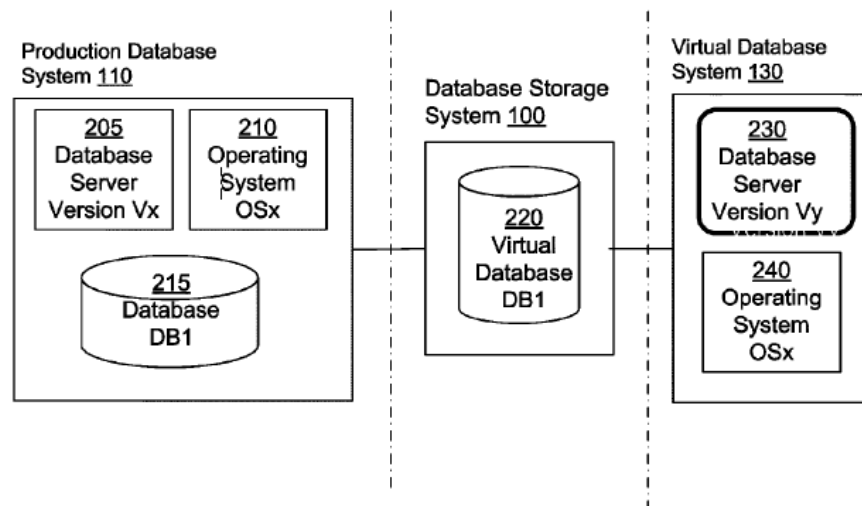


FIG. 2(a)

Figure 2a depicts production database system 110, virtual database DB1 220 stored in database storage system 100, and virtual database system 130, which accesses virtual database 220. *Id.* at 6:59–7:20.

To virtualize a production database, the system of the '808 patent makes a first “point-in-time” (“PIT”) copy of the production database and stores an entire set of database blocks representing the production database

at that time in database storage system 100. *See id.* at 18:27–36, Fig. 10. Subsequent PIT copies involve incremental changes and copy “only the blocks that changed since the last PIT copy and may copy much less data compared to the first PIT copy.” *Id.* at 18:38–41. A virtual database (VDB) is created by creating virtual database (VDB) file structures comprising VDB blocks that point to different PIT database blocks. *See id.* at 18:27–55. Each time an updated PIT copy is received at database storage system 100 reflecting changes in the production database, the system updates the appropriate VDB blocks in a VDB file which are “implemented as pointers to the actual database block that stores the [updated] data.” *See id.* at 18:44–55.

The ’808 patent discloses several embodiments for making PIT copies of the production database including, by streaming data to the database storage system and, alternatively, using file sharing. In “the streaming embodiment,” which is depicted in Figures 4 and 5 and described in column 12, line 14 to column 14, line 67 of the ’808 patent (*see id.* at 3:29–37 (describing Figures 4 and 5 as “an embodiment of the invention”)), the production database system, upon receiving a request for data from the point-in-time copy manager of the database storage system (*id.* at 12:19–23), packages the production database data “into a format that can be processed by the point-in-time copy manager” (*id.* at 12:58–62) and builds the appropriately formatted data into a data stream that is sent to the point-in-time copy manager. *Id.* at 12:62–13:3. In some embodiments, “the production system library 385 includes code” in this stream for analyzing the structures of the files of the database or process metadata “associated with database blocks.” *Id.* at 13:11–16.

Upon receiving the data stream, the point-in-time copy manager processes the data stream to identify database blocks contained in it. *Id.* at 13:27–33. In the data stream, “[e]ach database block includes metadata” (*id.* at 13:33–34), which is used, for example, to “identify database block boundaries in the stream of data” (*id.* at 14:22–25). When saving a retrieved database block into a transferred or copied database file on the database storage system, the point-in-time copy manager “analyzes the database block metadata to map [] the database block to [the] database file and a location within the file.” *Id.* at 14:29–31.

In the “file sharing embodiment” for making PIT copies of a production database, transfer of production database data is achieved by “using a file sharing system similar to the file sharing system 120” (*id.* at 7:57–64), such as a network file system (NFS) (*id.* at 10:35–37). Figure 1, which is identical in both the ’174 and the ’808 patents, is reproduced below.

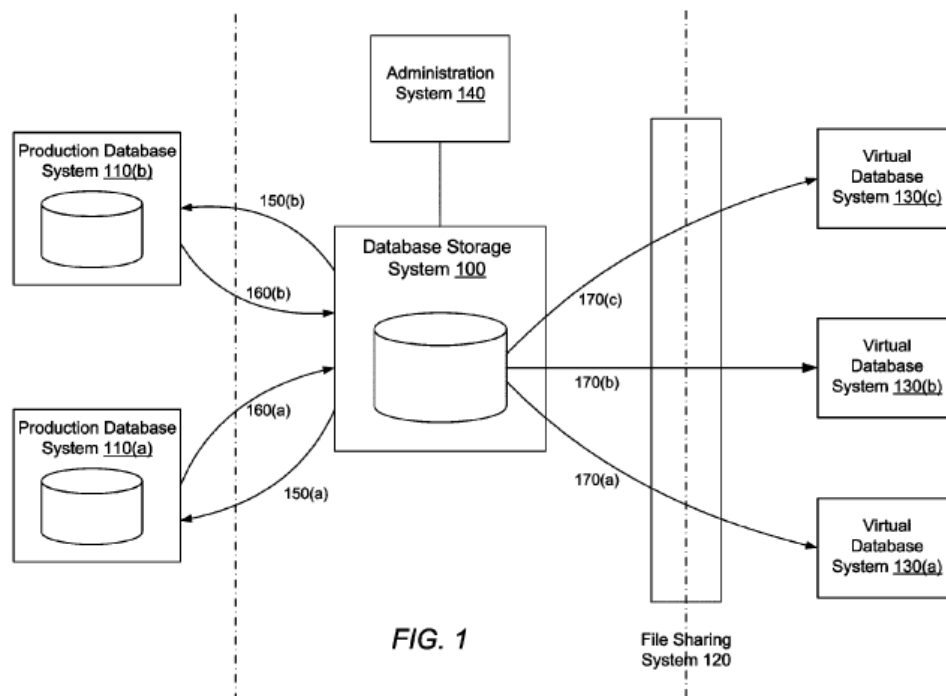


Figure 1 illustrates the file sharing embodiment for copying information from a production database to a database storage system to create a VDB. Ex. 1001, 4:11–14. Database storage system 100 periodically sends request 150 for data to a production database system 110. *Id.* at 4:26–35. Production database system 110 responds by sending information representing changes of data stored in the production database since the last response 160 sent by the production database system 110. *Id.* Database storage system 100 receives and stores the data and “may analyze the data 160 to determine whether to store the information or skip the information if the information is not useful for reconstructing the database at previous time points.” *Id.* at 4:35–41. “In some embodiments information may be copied from storage level snapshots of production databases or clones of production databases instead of a live production database.” *Id.* at 4:16–19.

Figure 10 from the '808 patent is reproduced below.

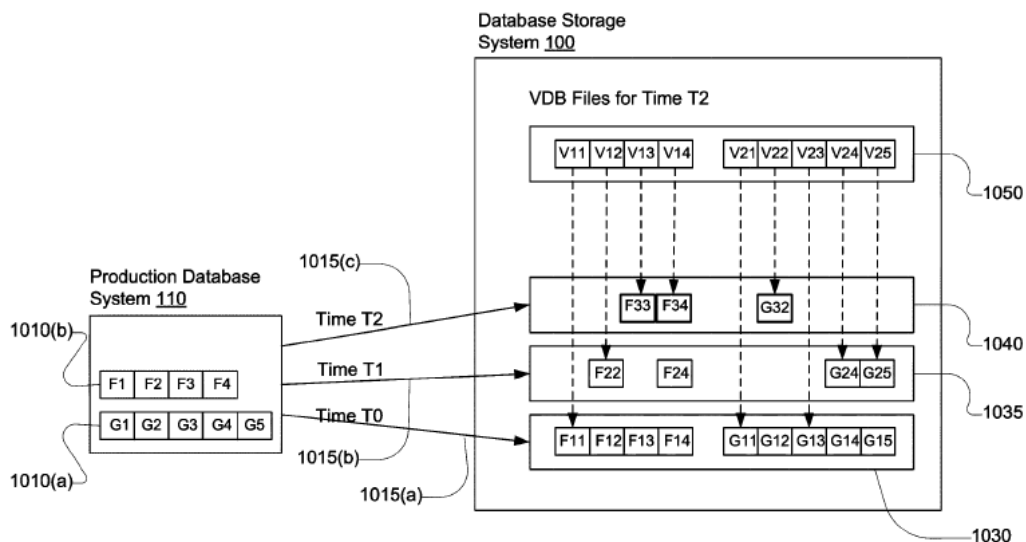


FIG. 10

Figure 10 shows “VDB Files for Time T2” in database storage system 100. Figure 10 further shows that “VDB file structures 1050” include blocks V11, V12, V13, and V14 which point to database blocks F11 . . . F34 that represent different PIT (i.e., at times T0, T1, and T2) copies of production database blocks F1, F2, F3, and F4 at production database system 110. Initially, all the production database blocks are copied to create “[t]he first PIT copy 1030 made at time T0,” as represented by database blocks F11, F12, F13, and F14 in database storage system 100. Ex. 2002, 18:35–38. Later, when the PIT copy made at time T2 is received and the VDB blocks are updated, block V13 points to the updated data at block F33, which represents a change existing at T2 to the data in block F3 in the production database (*see id.* at 18:53–55), whereas VDB block V11 still points to the data in block F11 “since the [production database] block F1 was never updated during copies made at time T1 and T2” (*id.* at 18:49–51).

Figure 9 of the ’174 patent illustrates how database blocks may be shared by file structured created for different VDBs. Figure 9 of the ’174 patent is reproduced below.

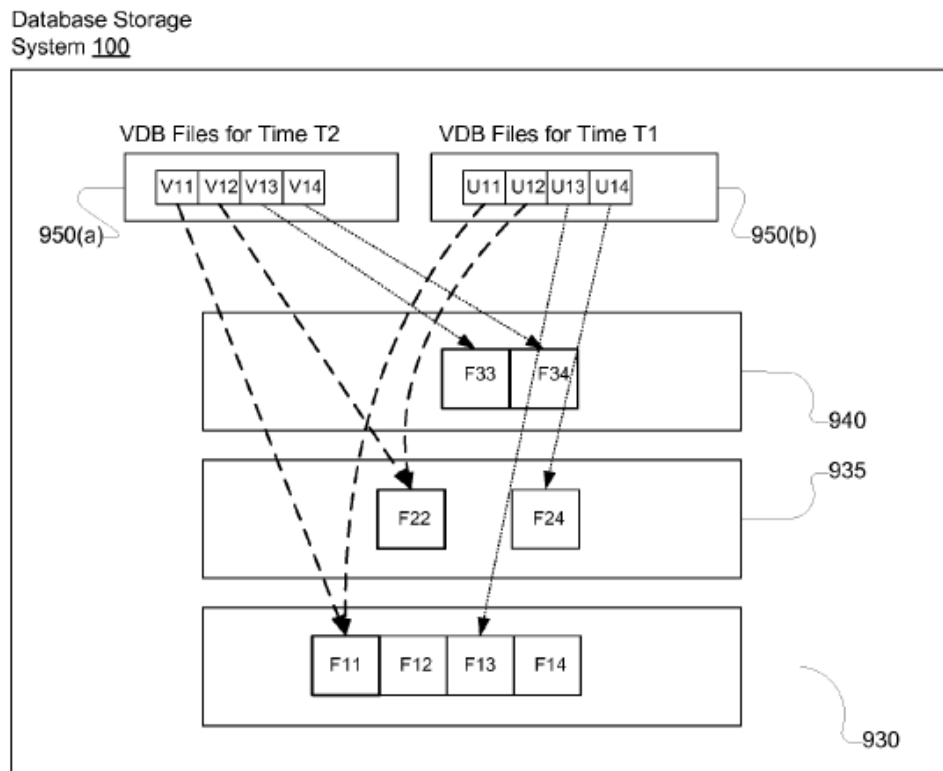


FIG. 9

Figure 9 of the '174 patent illustrates how database blocks stored on the storage system data store may be shared by file structures created for different virtual databases. Ex. 1001, 2:28–31, 14:1–3. Blocks V13 and V14 of the file structure 950(a) point at the latest copy of the blocks F33 and F34 that are not shared with the VDB files 950(b). *Id.* at 14:4–6. However, Block V11 of file structure 950(a) shares database block F11 with block U11 of file structure 950(b). *Id.* at 14:6–8. Similarly, Block V12 of file structure 950(a) shares database block F22 with block U12 of file structure 950(b). *Id.* at 14:8–9.

As mentioned above, the '174 patent incorporates by reference the entirety of the '808 patent for describing the creation and use of virtual databases, but the focus of the '174 patent is on *interfacing and interacting* with these systems. Ex. 1001, Title, 1:14–16, 4:6–9. To this end, the '174

patent discloses a user interface which “allows a database administrator to perform various actions supported by the database storage system.” *Id.* at 4:22–25. Figure 4 of the ’174 patent is reproduced below.

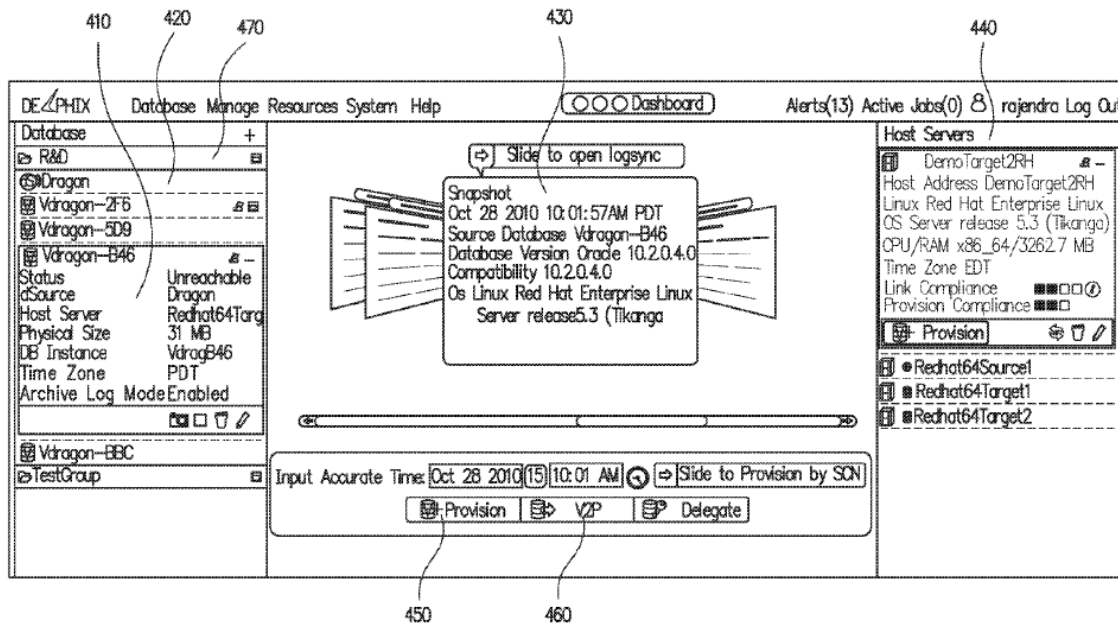


FIG. 4

Figure 4 of the ’174 patent illustrates an interface for allowing a user to select a source database to create a virtual database. *Id.* at 2:10–11. In the user interface, production database systems 110 are displayed as data sources 420 and “[e]ach data source [420] may display one or more source databases 410.” *Id.* at 11:21–26. “In response to the user selecting a particular source database 410, the user interface provides information describing the source database including its status, size, name etc. as well as information describing the various point-in-time copies 430 stored on the database storage system 100.” *Id.* at 11:35–39. For each point-in-time-copy 430, the information provided includes the time at which it was made, the source database from which it was made—which “can be a virtual database associated with the database source”—as well as information describing the

production database system 110 hosting the source database. *Id.* at 11:39–46. Figure 5 of the '174 patent is reproduced below.

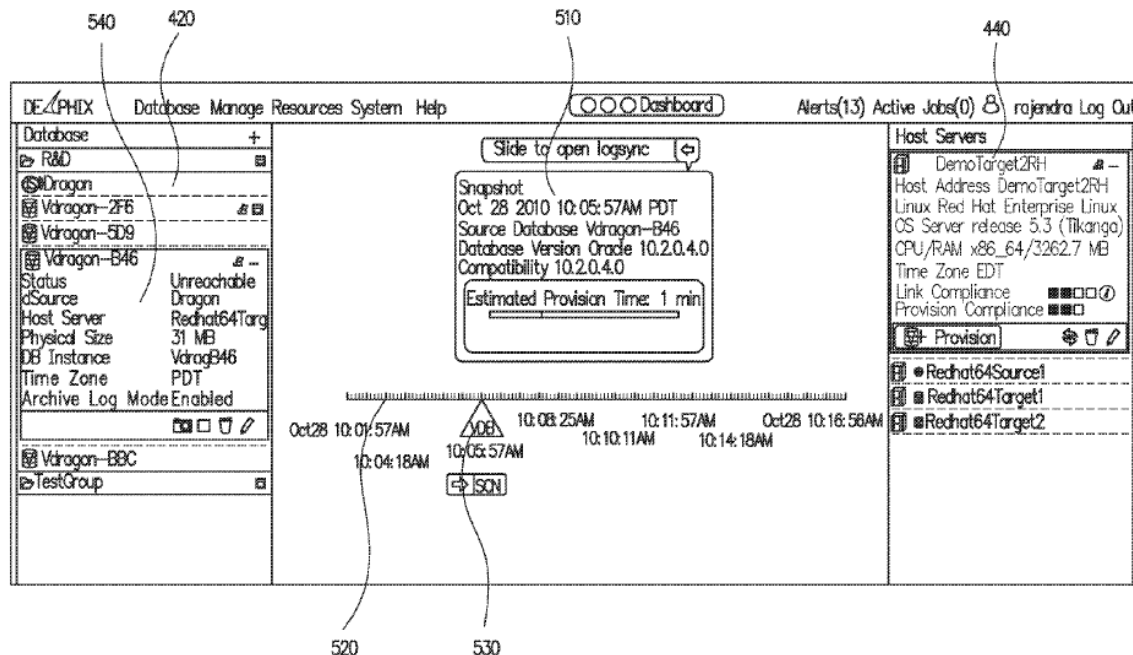


FIG. 5

Figure 5 of the '174 patent illustrates the interface for allowing a user to select a particular point in time associated with the source database for creating a virtual database based on the selected point in time. *Id.* at 2:13–16. In this example, the user selects a point-in-time copy 510 based on source database 540 for creation of a virtual database. *Id.* at 12:12–14. User interface 520, shown by a timeline, allows the user to select a time point, using slider 530, in between the time of copying of two point-in-time copies. *Id.* at 12:14–35.

D. Illustrative Claim

Of the challenged claims in the '174 patent, claims 1, 27, and 28 are independent. Claim 1 is illustrative and recites:

1. A method of creating a virtual database system, the method comprising:

receiving information identifying a source database for providing data for storage in a first virtual database and a second virtual database being created;

receiving information identifying a first point in time, such that the source database comprises information stored prior to the first point in time;

receiving information identifying a first destination database server for accessing the first virtual database being created; and

creating the first virtual database on a storage system storing a plurality of point-in-time copies of the source database, the point-in-time copies comprising database blocks such that at least some of the database blocks are associated with multiple point-in-time copies of the source database, the creating of the first virtual database comprising;

creating a first set of files for the first virtual database, each file in the first set of files linked with a first set of database blocks comprising information stored in the source database prior to the first point in time; and

mounting the first set of files on the first destination database server allowing the first destination database server to read from and write to the first set of files;

receiving information identifying a second point in time, such that the source database comprises information stored prior to the second point in time;

receiving information identifying a second destination database server for accessing the second virtual database being created; and

creating the second virtual database on the storage system,
the creating of the second virtual database comprising;

creating a second set of files for the second virtual
database, each file in the second set of files linked with a
second set of database blocks comprising information
stored in the source database prior to the second point in
time; and

mounting the second set of files on the second destination
database server allowing the second destination database
server to read from and write to the second set of files.

Ex. 1001, 18:57–19:31.

ANALYSIS

A. Printed Publication—Neto and Klivansky

Patent Owner contests that Neto and Klivansky are prior art “printed publications” in accordance with 35 U.S.C. §§ 102 and 311(b). Paper 17 (“PO Resp.”), 1–4. We look to the underlying facts to make a legal determination as to whether a document is a printed publication. *Suffolk Techs., LLC v. AOL Inc.*, 752 F.3d 1358, 1364 (Fed. Cir. 2014). The determination of whether a document is a “printed publication” under 35 U.S.C. § 102(b) involves a case-by-case inquiry into the facts and circumstances surrounding its disclosure to members of the public. *In re Klopfenstein*, 380 F.3d 1345, 1350 (Fed. Cir. 2004). Public accessibility is a key question in determining whether a document is a printed publication and is determined on a case-by-case basis. *Suffolk Techs.*, 752 F.3d at 1364. To qualify as a printed publication, a document “must have been sufficiently accessible to the public interested in the art.” *In re Lister*, 583 F.3d 1307, 1311 (Fed. Cir. 2009).

Initially, we note our disagreement with Patent Owner’s contention that Petitioner cannot rely upon evidence not submitted with the Petition to show that Neto and Klivansky are prior art. PO Resp. 2–4. In Patent Owner’s view, Petitioner must make out a *prima facie* case of unpatentability in its Petition, which includes the substantive element of Neto and Klivansky being publicly accessible and prior art. *Id.* at 3–4. That position, however, is not informed by the difference between the threshold for instituting a trial (35 U.S.C. § 314(a)) and proving unpatentability of a claim in trial (35 U.S.C. § 316(e)). As noted by our reviewing court, “there is a significant difference between a petitioner’s burden to establish a ‘reasonable likelihood of success’ at institution, and actually proving invalidity by a preponderance of the evidence at trial.” *TriVascular, Inc. v. Samuels*, 812 F.3d 1056, 1068 (Fed. Cir. 2016) (quoting 35 U.S.C. § 314(a) and comparing § 316(e)).

Based on the information presented in the Petition and Patent Owner’s Preliminary Response (Paper 6) we determined there was a reasonable likelihood that Petitioner would prevail in its challenges that included Neto and Klivansky. Inst. Dec. 31; *see* 35 U.S.C. § 314(a) (threshold for instituting *inter partes* review); *see also* 37 C.F.R. § 42.108(c) (“The Board’s decision [on Institution] will take into account a patent owner preliminary response where such a response is filed.”). Patent Owner did not challenge the prior art status of any of the applied patents or publications in its Preliminary Response. Patent Owner, in fact, stated that it had “disclosed to the Patent Office every NetApp feature that Petitioner now cites in the Petition” and that Edwards, Edwards II, Neto, and Klivansky “were *published* over a span of five years.” Paper 6, 45, 47 (emphasis

added). We do not mean to suggest that a patent owner must raise any “printed publication” issues in a preliminary response in order for the Board to consider such issues in the preliminary proceeding phase. In this case, however, based in part on the information in Patent Owner’s Preliminary Response and in part on the printed dates and the lack of indicia of confidentiality or internal, non-public distribution in Neto and Klivansky, we determined that Petitioner had met its burden for a threshold showing to proceed to trial.

Patent Owner also argues that Petitioner cannot rely on declarations filed after the Petition. These declarations include two declarations provided by Louis Hernandez (Ex. 1020) and Joseph Ortiz (Ex. 1028) in response to objections by Patent Owner⁹ and a Supplemental Declaration by Mr. Hernandez (Ex. 1048) filed with its Reply. PO Resp. 2–4. Our rules authorize serving supplemental evidence in response to an objection. 37 C.F.R. § 42.64(b)(2). Patent Owner lacks a foundation to complain that evidence has been produced in response to its objections. Petitioner also relies, properly, on the supplemental evidence in its Reply, as evidence in reply to Patent Owner’s arguments in its Response that Neto and Klivansky are not printed publications.

Turning to the substance of Exhibit 1020, Mr. Hernandez testifies that he is currently employed by Petitioner, was employed by NetApp from 2004 to 2009, and was a NetApp customer from 2000 to 2004. Ex. 1020 ¶¶ 1, 2, 4. Mr. Hernandez testifies further that for most of his time at NetApp, as a

⁹ Exhibit 1028 is expunged at Petitioner’s request. We do not further discuss the Exhibit. We address Patent Owner’s motions to exclude these documents in a separate section, below.

Systems Engineer, he was responsible for marketing NetApp's products and services to numerous customers, prospective customers, business partners, and/or alliances. *Id.* ¶ 3. "During the 2000-2009 time-frame, to support its marketing efforts, it was NetApp's standard practice to publish technical reports, white papers, and product manuals or guides to customers, potential customers, business partners, and alliances." *Id.* ¶ 6. "These documents were published, according to standard practice, as of the month and year that appeared on the face of the documents." *Id.* Mr. Hernandez testifies that he has personal knowledge of and recognizes Neto and Klivansky, and that they were published during his tenure at NetApp or his subsequent tenure at Midwave, a certified distributor and reseller of NetApp. *Id.* ¶¶ 7, 8, 14–17.

Patent Owner argues Mr. Hernandez does not declare that Neto or Klivansky was "publicly accessible." PO Resp. 2–3. Patent Owner submits:

Even if it was NetApp's "standard practice" to provide its documents to its "customers, potential customers, business partners and alliances," that does not establish that these documents were available to *the public*, but instead shows at most that they were only available to a subset of entities affiliated with NetApp.

Id. at 3.

Petitioner replies with a Supplemental Declaration from Mr. Hernandez. Paper 24 ("Reply") 7–8 (citing Ex. 1048). Mr. Hernandez testifies that he uses the term "publish" or "published" as referring to documents being publicly distributed to customers, potential customers, business partners, and alliances as of the month and year that appeared on the face of the documents, non-confidentially. Ex. 1048 ¶ 5. According to Mr. Hernandez, NetApp had more than two hundred Systems Engineers and

other sales personnel during the relevant timeframe (*id.* ¶ 4) and that technical reports, white papers, product manuals, and product guides were freely distributed to support its marketing efforts (*id.* ¶ 7). Mr. Hernandez testifies further that it was important for NetApp to date the documents accurately so that customers and potential customers could understand if a specific document accurately reflected features for specific versions of NetApp’s products or if a document was outdated or updated to reflect more current features. *Id.* ¶ 10. Further, Petitioner provides evidence that by 2007 there were more than 94,000 NetApp systems deployed and the company had thousands of customers in 138 countries. Reply 6; Ex. 1058, 3.¹⁰

As part of routine discovery (37 C.F.R. § 42.51(b)(1)(ii)), Patent Owner had the opportunity to cross-examine Mr. Hernandez during Patent Owner’s first discovery period but elected not to. Patent Owner cross-examined Mr. Hernandez in its second discovery period regarding the testimony in his Supplemental Declaration. We have considered Patent Owner’s Motion for Observation on Cross-Examination Testimony of Mr. Hernandez (Paper 38) and Petitioner’s Response (Paper 44), insofar as they relate to public accessibility of Neto and Klivansky.¹¹ We acknowledge the potential for bias in Mr. Hernandez’s testimony as a present employee of

¹⁰ We find that Exhibit 1058, a NetApp Form 10-K SEC filing, was properly submitted by Petitioner as evidence in rebuttal to Patent Owner’s public accessibility challenge in its Response.

¹¹ During the Hearing, Patent Owner asked for, and we granted, additional time to consider its oral Hearing arguments regarding alleged new issues (regarding publication) raised in Petitioner’s Reply in lieu of filing a Sur-Reply. See Paper 60, 211:16–212:18; 224:13–21; 237:1–25.

Petitioner. We find, however, the testimony in Mr. Hernandez's Declarations as to public accessibility of Neto and Klivansky to be credible. As an earlier panel of the Board has found, in a proceeding involving a different patent and different parties, documents such as Neto and Klivansky are dated technical documents or whitepapers, having no indication of being mere drafts or internal papers, each of which is "a type of document whose very purpose is public disclosure." *Veeam Sw. Corp. v. Symantec Corp.*, Case IPR2014-00089, slip op. at 14 (PTAB Apr. 25, 2014) (Paper 9).

Finally, Petitioner also submits a declaration it says was produced in response to Patent Owner's evidentiary objections. Reply 5. Petitioner provided the declaration from the office manager of the Internet Archive (Wayback Machine). *Id.* (citing Ex. 1023). Klivansky is dated October 2004 and is designated "TR3347" (or Technical Report 3347). Ex. 1006, 1. According to the testimony regarding how the Internet Archive works (Ex. 1023 ¶¶ 3–5), we find the evidence indicates that Klivansky was available on NetApp's commercial website on, or at least as early as, November 17, 2004. Ex. 1023, 38. Exhibit 1023 indicates that Klivansky was, thus, "sufficiently accessible to the public interested in the art." *See In re Lister*, 583 F.3d at 1311.

"A given reference is 'publicly accessible' upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence, can locate it." *SRI Int'l, Inc. v. Internet Sec. Sys., Inc.* 511 F.3d 1186, 1194 (Fed. Cir. 2008) (quoting *Bruckelmyer v. Ground Heaters, Inc.*, 445 F.3d 1374, 1378 (Fed. Cir. 2006)). Petitioner asserts that the level of ordinary skill is not disputed

in this case and that “[t]here is no difference between customers and potential customers of NetApp and a [person of ordinary skill in the art].” Reply 6–7. Mr. Hernandez testifies that NetApp’s customers and potential customers were entities or individuals who were “interested in data storage technology, data management technology, database storage and management technology, and related technologies.” Ex. 1048 ¶ 3. As discussed above, Mr. Hernandez also testifies NetApp had more than two hundred Systems Engineers and other sales personnel during the period of 2004–2009 who distributed NetApp’s technical documents (including Neto) to “thousands” of customers and potential customers. *Id.* ¶ 4. Hence, the record supports Petitioner’s contention that Neto was distributed to persons interested and ordinarily skilled in the subject matter of database technology at the time of its publication. See Reply 6–7. In view of the foregoing considerations, we find that Petitioner has established, by a preponderance of the evidence, that Neto (dated April 2009) was sufficiently disseminated to persons of ordinary skill interested in database technology to be deemed “publicly accessible” at least more than one year before November 30, 2010, the earliest possible priority date of the ’174 patent. See Ex. 1001, (60). Petitioner has also established, by a preponderance of the evidence, that Klivansky (dated October 2004) was available to the interested public at least more than one year before the critical date. Therefore, on this record, we determine Neto and Klivansky qualify as prior art printed publications under 35 U.S.C. § 102(b).

B. Claim Construction

We interpret claims of an unexpired patent using the broadest reasonable construction in light of the specification of the patent in which

they appear. *See* 37 C.F.R. § 42.100(b). On this record and for purposes of this Decision, we determine that only the claim terms addressed below require express construction.

1. “*database block*”

In the Institution Decision, we preliminarily construed the term “database block,” as “a unit of data used by a database.” Inst. Dec. 14. In its Response (Paper 17, “PO Resp.”), Patent Owner asserts that this definition is impermissibly broad. PO Resp. 18–28. Patent Owner’s proposed construction is “a unit of data used by a database *which comprises a specific number of bytes stored in the storage, a portion of which stores metadata associated with the unit of data.*” *Id.* at 18 (emphasis added). Petitioner, on the other hand, agrees with our conclusion in the Decision to Institute. Paper 25 (“Reply”) 11–15.

a. *Metadata Associated with the Database Block*

The main dispute between the parties centers on whether a database block must necessarily include metadata. Patent Owner asserts that it does (PO Resp. 18), and Petitioner disagrees (Pet. 11). We agree with Petitioner.

We begin our analysis by considering the language of the claims themselves. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1314, 1315 (Fed. Cir. 2005) (*en banc*). The term “metadata” is not recited in any of the claims of the ’174 patent. The ’808 patent, however, includes two claims that recite “metadata”—dependent claims 32 and 33, which are not challenged in any proceeding, of which we are aware, currently before the Board. These claims depend indirectly from claim 1 of the ’808 patent and expressly recite “metadata of database blocks.” Thus, had the patentees intended to limit “database blocks” recited in the claims of the ’174 patent to require

metadata, they demonstratively could have done so by explicitly modifying the disputed term with “metadata,” but did not.

Moving to the specification, Patent Owner asserts that the following passage in the Summary section of the ’808 patent defines the term “database block.”

A database block is a unit of data used by a database and comprises a specific number of bytes stored in the storage. A database block can also be referred to as a page. A portion of the database block stores metadata associated with the database block.

PO Resp. 18–19 (quoting Ex. 2002, 2:7–12).

The first phrase in the cited passage above explicitly defines the term, by stating “[a] database block is a unit of data used by a database.”¹² Ex. 2002, 2:7–9. Although the sentence following shortly thereafter states that a database block “stores metadata,” that sentence by itself is insufficient to limit the disputed term by requiring the unclaimed “metadata” feature because it does not state unambiguously that all “database blocks” must include metadata. *See Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 906 (Fed. Cir. 2004) (construing a claim term broadly because “[n]o statement in the written description [] constitute[d] a limitation on the scope of the invention”) (quoting *Brookhill-Wilk 1, LLC v. Intuitive Surgical, Inc.*, 334 F.3d 1294, 1301 (Fed. Cir. 2003)). Further, the cited passage also does

¹² The first sentence also states a database block comprises “a specific number of bytes stored in the storage.” For the reasons discussed below, we find this addition is not part of the explicit definition but, rather, represents embodiments within the defined term. Patent Owner also fails to explain how to interpret this particular phrase or how it presents a material issue related to the prior art.

not exclude the possibility of some database blocks not having any metadata. *See id.* at 908 (passages in the Summary of the Invention section of a patent did not limit the scope of the invention because the passages, “although focusing on the use of the invention in conjunction with pressure jackets, do not disclaim the use of the invention in the absence of a pressure jacket”). More importantly, the rest of the ’174 patent is consistent with a definition of database blocks that does not require metadata in all database blocks.

Patent Owner asserts, citing certain portions of the ’808 patent and the Declaration of Prashant Shenoy, Ph.D. (Ex. 2030, “Shenoy Decl.”), that if a database block does not include metadata, the disclosed system would not work as described. PO Resp. 19–20. For example, Patent Owner argues that a database block must include metadata because the disclosed system analyzes the metadata of each block to store only the incremental changes made to the production database (*id.* (citing Ex. 2002, 13:34–36, 46–51)), which is “one of the main functions” of the claimed system (*id.* at 19 (citing Ex. 2002, 6:30–34, 40–43, 7:49–57; Ex. 2030 ¶¶ 85–86)) essential to achieving “a main purpose” of the invention—“to efficiently provide virtual databases . . . without proliferating redundant copies of database data” (*id.* (citing Ex. 2030 ¶ 86)). Patent Owner also asserts that metadata is required in each database block in order to map the block to a database file and a location within that file. *Id.* at 20 (citing Ex. 2002, 14:27–31).

Patent Owner’s argument is unpersuasive because the argued advantages or purposes are not recited features of the claims. Moreover, a claim is not required to encompass all of the advantages or purposes of the invention. *See Howmedica Osteonics Corp. v. Wright Med. Tech., Inc.*, 540 F.3d 1337, 1345 (Fed. Cir. 2008) (“An invention may possess a number of

advantages or purposes, and there is no requirement that every claim directed to that invention be limited to encompass all of them.”) (citations omitted).

Moreover, Patent Owner focuses its arguments on the streaming embodiment, however, Patent Owner does not show that the file sharing embodiment requires this approach, where the database files to be copied have defined boundaries and known structures, and the database blocks stored in the files can be accessed directly. *See, e.g.*, Ex. 2002, 13:10–12 (“the production system library [] includes code to analyze *the structures of the files* of the database stored in the data store and *also includes code to process metadata associated with database blocks* stored in the data store”) (emphases added),¹³ 6:11–17 (“FIG. 1 illustrates one embodiment for how information may be *copied* from a production database to a database storage system . . . *using a file sharing system*. . . . In some embodiments information may be *copied* from *storage level snapshots* of production databases.”) (emphases added). When file sharing, as opposed to streaming, is used, the database file on the production system can be accessed and copied directly by “mounting the production DB data store” on the database storage system without packing and unpacking the database blocks of the database file into and out of data streams. Ex. 1001, 5:29–36.

Considering “the context of the surrounding words” to the term “database block” in the claims, which “must be considered in determining the ordinary and customary meaning” of the disputed term, *ACTV, Inc. v.*

¹³ The phrase “metadata *associated* with database blocks” implies any metadata need not be *in* the database blocks.

Walt Disney Co., 346 F.3d 1082, 1088 (Fed. Cir. 2003), independent claims 1, 27, and 28 recite creating a virtual database on a storage system storing a plurality of point-in-time copies of the source database. But the claims do not say anything about a particular method of transferring point-in-time copies, whether by streaming or by file sharing. Hence, the claims are not limited to either embodiment, and, therefore, the streaming embodiment cited by Patent Owner does not limit the claims.

Further, even looking at the streaming embodiment, we are not persuaded that metadata is required by database blocks. Patent Owner points only to a single streaming embodiment (PO Resp. 19–20), in which the data stream may include unnecessary database blocks, such as the blocks that did not change since the last point-in-time copy was transmitted, which are eliminated after the data stream is received at the database storage system by analyzing metadata for each database block. Ex. 2002, 13:43–64. In an alternative embodiment, which is not addressed by Patent Owner, the unchanged blocks are eliminated at the production system and never sent to the database storage system. *Id.* at 13:64–14:3 (“In *other* [sic] *embodiment*, some or all of the unnecessary blocks may be *eliminated while the data stream is built* by the production system library []. In this embodiment, the data stream . . . is reduced in size resulting in efficient communication between” the production system and the database storage system (emphases added)). Patent Owner does not explain why metadata must be included in each database block to achieve the incremental update function in this second embodiment. Hence, packing metadata within database blocks may be involved in *some* streaming embodiments, but nothing in the specification indicates it is required for the incremental update function. Therefore,

nothing in the specification indicates that copying database files by streaming data is the essence of the claimed invention rather than a preferred embodiment, which may not be read into the claims “absent clear disclaimer in the specification.” *In re Am. Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1369 (Fed. Cir. 2004).

Therefore, in view of the entire disclosure of the ’174 patent and the plain language of the claims, we find that the intrinsic record before us does not justify limiting the term “database block” by reading in the “metadata” limitation not found in the claims. *See Liebel-Flarsheim Co.*, 358 F.3d at 908.

Patent Owner also argues additional evidence supports its proposed construction requiring data blocks to include metadata. For example, citing the testimony of Dr. Shenoy, Patent Owner asserts that all database management systems mentioned in the ’174 patent, such as Oracle and IBM DB2 (Ex. 1001, 3:11–15), require metadata in database blocks. PO Resp. 21–22 (citing Ex. 2030 ¶¶ 41–46, 83–84). In the paragraphs cited by Patent Owner, Dr. Shenoy discusses various documents describing the database systems listed in the ’174 patent, including Oracle, Sybase, Microsoft SQL Server, and IBM DB2, and testifies that these database systems all require metadata in database blocks.¹⁴ Ex. 2030 ¶¶ 43–46, 83–84 (citing testimony from Petitioner’s declarant, Dr. Zadok in another related case). Relying upon the testimony of Dr. Shenoy, Patent Owner also asserts that “[i]t is well known to one skilled in the art that a database block necessarily includes

¹⁴ However, neither Patent Owner’s brief nor Dr. Shenoy’s testimony discuss the MYSQL database system mentioned in the ’174 patent.

metadata.” PO Resp. 21 (citing Ex. 2030 ¶¶ 83–84); *see also id.* at 19 (“[T]hat database blocks include metadata is consistent with the ordinary meaning of the term in the database field and comports with how the term is used by every major database system provider.”) (citing Ex. 2030 ¶¶ 41–46, 83). In his Declaration, Dr. Shenoy discusses a treatise on database systems (Ex. 2015, “Molina”) and testifies that it is generally understood that a database block or a page will include metadata. Ex. 2030 ¶ 41–42 (citing Ex. 2015, 29, 31).

The evidence presented by Patent Owner in support of its argument—i.e., testimony of experts and documents describing the commercially available database systems listed in the ’174 patent—is properly characterized as extrinsic evidence. Such extrinsic evidence is “less significant than the intrinsic record in determining the legally operative meaning of claim language.” *Phillips*, 415 F.3d at 1317 (citations omitted) (internal quotation marks omitted).

Petitioner argues that Patent Owner does not provide any evidence that database blocks require metadata in database systems other than relational database systems. *See* Reply 13. Petitioner correctly points out that the ’174 patent specification states that the disclosed invention “can be used for *any database*.” *Id.* (quoting Ex. 1001, 3:15–19) (emphasis by Petitioner). Further, the ’808 patent states that “[a] database may be implemented using a database model, for example, a relational mode, object model, hierarchical mode or network model” and goes on to state that “the techniques disclosed can be used for *any database*.” Ex. 2002, 5:8–15. Thus, we agree with Petitioner that the ’174 patent does not limit its disclosed invention to relational database technology.

Similarly, the portion of the Molina treatise relied upon by Dr. Shenoy appears to describe features of a *relational* database system, not characteristics common to all databases in general. *See, e.g.*, Ex. 2015, 29 (“Collections such as *relations* are usually represented by placing the records that represent their data elements in one or more blocks.”) (emphasis added), 31 (“Records representing tuples of *a relation* are stored in blocks of the disk . . . there is a *block header* holding information such as: . . . Information about which *relation* the tuples of this block belong to.”) (first and last emphases added).

Relying on the testimony of Dr. Zadok submitted with its Reply (Ex. 1064), Petitioner asserts that, in other type of database systems, such as Google’s BigTable database, metadata is stored separately from the database blocks.¹⁵ Reply 13 (citing Ex. 1049, 4; Ex. 1064 ¶ 25). Petitioner also argues, citing the testimony of Dr. Zadok and another treatise on database systems (Ex. 1050, “Elmasri”), that a “flat file database,” such as a comma separated value (CSV) file used by spreadsheet applications, does not have metadata in its blocks. Reply 13 (citing Ex. 1064 ¶¶ 26–29; Ex. 1050, 431;

¹⁵ Petitioner argues the definition of a “database” proposed by Patent Owner’s expert, “a collection of data that is organized so that it can be easily accessed, managed or updated,” encompasses many types of databases other than the relational databases. *See* Reply 13 (citing Ex. 2030 ¶ 33). In addition to disclosing that “the techniques disclosed can be used for any database,” (Ex. 1001, 5:15–19) the ’808 patent provides another broad description— “[a] database comprises data stored in a computer for use by computer implemented applications” (Ex. 2002, 4:67–5:1). Under either description or Mr. Shenoy’s testimony, we agree with Petitioner that a “database” encompasses many types of databases other than the relational databases, including Google’s BigTable, flatfiles, and spreadsheet databases, and “any database.”

Ex. 1053 (flat file database “is unlike a relational database” according to techopedia.com); Ex. 1055, 15 (examples of flat files or ordinary text files)). Therefore, Petitioner asserts that the ordinary meaning of database block does not require including metadata. *See id.*, Ex. 1064 ¶ 22.

We agree with Petitioner that all extrinsic evidence Patent Owner relies upon to argue ordinary meaning appears to be directed to relational databases. Because the challenged claims plainly recite “database,” and, therefore, are not limited to relational databases, Patent Owner’s evidence, even assuming it shows relational databases all have database blocks that require metadata, does not establish an ordinary and customary meaning of “database blocks” recited in the claims. Therefore, we find Patent Owner’s extrinsic evidence regarding ordinary meaning does not overcome the intrinsic record of this case. *See Phillips*, 415 F.3d at 1318 (“[A] court should discount any expert testimony that is clearly at odds with the claim construction mandated by the claims themselves, the written description, and the prosecution history, in other words, with the written record of the patent.”) (citations and internal quotation marks omitted). Furthermore, we are persuaded by Petitioner’s evidence that certain well-known databases, such as Google’s BigTable database, do not require metadata in database blocks, and, therefore, find that the evidence in this case does not establish a widely accepted meaning of “database block” in the field of database systems that requires metadata to be necessarily included in a database block.

b. “Application Level” v. “Storage Level”

Patent Owner also argues, citing the passages in the ’808 patent describing the streaming embodiment, that a “database block” must include

metadata because “the ’174 patent describes an *application level* system that understands database blocks and uses metadata in those database blocks to determine whether to store database blocks and where to store them.” PO Resp. 28 (emphasis added); *see also id.* at 16–18 (making the same argument) (citing Ex. 1001, 4:37–41; Ex. 2002, 13:43–60, 14:27–41). Patent Owner asserts, therefore, a “database block” cannot be a “*storage level* unit of data.” *Id.* at 25 (emphasis added). The passages of the ’808 patent cited by Patent Owner, however, describe solely the streaming embodiment and, therefore, do not limit the challenged claims as discussed above.¹⁶

Patent Owner also asserts that the system of the ’174 patent “operates at the application level” because the system uses APIs (application program interfaces) to copy database blocks. *Id.* at 16–17 (citing Ex. 1001, 5:19–24; Ex. 2030 ¶ 72). Petitioner argues that Patent Owner’s argument is contrary to disclosed embodiments of the ’174 patent because the patent teaches the system “may retrieve the necessary database blocks from *storage level snapshots* of production databases.” Reply 14 (citing Ex. 1001, 6:4–8) (emphasis added by Petitioner). We agree with Petitioner’s argument and further note that the ’174 patent also discloses copying of database blocks from *storage level* snapshots of a database. *See* Ex. 1001, 4:11–19 (“FIG. 1

¹⁶ Patent Owner also cites the ’174 patent’s statement (PO Resp. 17) that “[t]he database storage system 100 *may* analyze the data 160 received to determine whether to store the information or skip the information if the information is not useful for reconstructing the database at previous time points.” Ex. 1001, 4:37–41 (emphasis added). This statement, however, uses permissive language and does not limit the claims as asserted by Patent Owner.

illustrates one embodiment for how information may be *copied* from a production database to a database storage system . . . using a file sharing system. . . . In some embodiments information may be *copied* from *storage level snapshots* of production databases.”) (emphases added). More importantly, Patent Owner does not explain why the fact that the system of the ’174 patent uses APIs to access database blocks necessarily requires the database blocks to include metadata. For example, Patent Owner does not identify, nor do we discern, anything in the written description that explains why the APIs cannot access and process metadata that is stored separately from database blocks.

Patent Owner further relies upon the testimony of Dr. Shenoy and external documents to argue that a “database block” is an “application level” entity, which is different from a “file system block,” an “operating system block,” or a “storage level” unit of data. PO Resp. 25–28 (citing Ex. 2030 ¶¶ 27–33). In particular, Patent Owner asserts that an Oracle document (Ex. 2004, “Oracle Database Concepts 11g Release 2 (11.2)”) explains this “well-known distinction.” *Id.* at 25–26 (citing Ex. 2004, 250, Fig. 12-5). Patent Owner further argues “neither are file system blocks accessible to a database nor are database blocks accessible to the file system.” *Id.* at 27 (citing Ex. 2030 ¶¶ 94–98). Therefore, Patent Owner argues a construction of the term “database block” that encompasses a storage unit block is incorrect. *Id.* at 28.

As a threshold matter, we note that the Oracle document is dated May, 2014, which is several years later than the filing date of the ’174 patent. Hence, the Oracle document is not probative of what was known to one of ordinary skill in the art at the time of the invention. More importantly, as

discussed above, we find the written description of the '174 patent does not support a limiting construction of the term “database block” based on the purported “application level” versus “storage level” distinction. Further, we do not find any disclosure in the specification that describes or discusses whether a “database block” is an “application level” or a “storage level” entity or construct. Hence, we find the testimony of Dr. Shenoy and the external documents discussed by Patent Owner regarding the “application level” versus “storage level” distinction to be largely divorced from the written description, and, therefore, insufficient to overcome the plain language of the claims. *See Kara Tech., Inc. v. Stamps.com, Inc.*, 582 F.3d 1341, 1348 (Fed. Cir. 2009); *Phillips*, 415 F.3d at 1318.

c. A Specific Number of Bytes Stored in the Storage

In the Institution Decision, we found that database blocks of the '174 patent need not have a specific number of bytes stored in storage because an empty database block would not have metadata or “a specific number of bytes stored in the storage.” Inst. Dec. 13–14. Patent Owner continues to argue that this is a requirement of the proper construction of “database blocks.” PO Resp. 18.

Patent Owner does not explain how to interpret “a specific number of bytes stored in storage.” For example, it is not clear if “a specific number” is a constant. In another argument, Patent Owner states that “[d]atabase blocks may be any number of different sizes independent of the size of the file system data blocks which may ultimately store the database data.” PO Resp. 27 (citing Ex. 2030 ¶ 95). Patent Owner also states that an IBM DB2 database page can be up to 32KB in size, spanning 8 WAFL data blocks,

most of which will not contain any metadata for the database page. *Id.* at 28 (citing Ex. 2017 at 128).

According to Patent Owner’s arguments, a database block may be “any number of different sizes.” Therefore, the database blocks of the ’077 patent need not have a constant specific number of bytes stored in storage.

d. “*Database Block*” v. “*File*”

Next, Patent Owner argues that the construction proposed by Petitioner (Pet. 9) and preliminarily adopted in our Institution Decision (Inst. Dec. 14)—“a unit of data used by a database”—is improper because it would equate a “database block” with a “file.” PO Resp. 23–25. Citing dictionaries, Patent Owner argues that the ordinary meaning of “file” is “a unit of data.” *Id.* at 24 (citing Ex. 2022, 3; Ex. 2023, 3). Patent Owner further asserts that our preliminary construction incorrectly encompasses a log file described in the Specification because a log file is also “a unit of data used by database.” *Id.*

Petitioner argues, citing the deposition testimony of Patent Owner’s expert, Dr. Shenoy, Patent Owner is incorrect because a file has a name associated with it but a database block, as construed by Petitioner, would not. Reply 14–15 (citing Ex. 1031, 261:19–20, 262:8–9). We agree with Petitioner that defining a database block as “a unit of data used by a database” does not equate a database block with a file.

First, a file is not necessarily “a unit of data used by a database” because not all files are used by a database. Conversely, the phrase “a unit of data used by a database” is not sufficient to describe a file because a file has additional properties or characteristics, such as a unique file name. In fact, the full definitions of a “file” provided in the Patent Owner cited

dictionaries, including the portions Patent Owner had omitted when quoting the sources in the Patent Owner Response, show that a file name is a required attribute of a file. *See* Ex. 2022, 3 (defining “file” as “[a] document or other collection of information stored on a disk and identified as a unit **by a unique name**”), Ex. 2023, 3 (defining “file” as “[a] collection of data or information that is stored as a unit in the computer under a single name, **called the file name**”) (the portions omitted by Patent Owner indicated with bold emphasis). Hence, the fact that a definition of a “database block” reads on some “files,” such as a log file used by a database, does not mean the definition equates the two terms because a file has additional properties or characteristics, such as a file name.

e. Conclusion

In summary, we find nothing in the intrinsic record, including the written description and the language of the claims, that justifies limiting the term “database block” by reading in the “metadata” or “specific number of bytes” limitations not found in the claims. Furthermore, no extrinsic evidence, including the testimony of experts, presented in this case is sufficient to overcome the plain claim language of the term “database block.” Therefore, on this record, we construe “database block” to mean “a unit of data used by a database.” *See Phillips*, 415 F.3d at 1316 (“The construction that stays true to the claim language and most naturally aligns with the patent’s description of the invention will be, in the end, the correct construction.”) (citation omitted).

2. “*virtual database*”

In the Institution Decision, we preliminarily construed the term “virtual database,” to mean “a set of database files capable of being read

from and written to, and capable of being mapped to physical addresses for stored database blocks.” Inst. Dec. 20. Petitioner agrees with this construction. Reply 15–18. Patent Owner, on the other hand, asserts that this definition reads out the “virtual” requirement. PO Resp. 31. Patent Owner, instead proposes that the term should be interpreted to mean “a set of files to which a database server can read and write such that the physical implementation of the database files is decoupled from the logical use of the database files by the database server.” *Id.* at 28.

To analyze Patent Owner’s proposed construction, Patent Owner defines a virtual database as “a set of files” or “database files” and modifies the definition with two phrases, each expressing a concept involving a database server: (1) “a database server can read and write” to the set of files, and (2) “the physical implementation of the database files is decoupled from the logical use of the database files by the database server.” We address each proposed modifying phrase in turn.

a. Database Server Can Read and Write

Patent Owner argues that the point of the first proposed modifying phrase is to “require something of the files . . . that they are of a form readable and writable by a database server.” PO Resp. 32. In other words, the main thrust of Patent Owner’s proposed phrase is that a database server can read from and write to a set of files.

Challenged independent claims 1, 27, and 28 specifically recite “allowing the . . . database server to read from and write to the . . . set of files [associated with the virtual database],” which encompasses essentially the same concept as Patent Owner’s proposed phrase. Hence, including the phrase “a database server can read and write [to a set of files]” in the

definition of the stand-alone term “virtual database” as an inherent attribute of the term would render the recited limitation superfluous. Such a construction is presumed improper. *See Digital-Vending Servs. Int’l, LLC v. Univ. of Phoenix, Inc.*, 672 F.3d 1270, 1274–75 (Fed. Cir. 2012) (rejecting the district court’s construction narrowing a term by a superfluous limitation when the claims explicitly recited the narrowing limitation, and discussing the “well-established rule that claims are interpreted with an eye toward giving effect to all terms in the claim”) (quoting *Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 950 (Fed. Cir. 2006)) (internal quotation marks omitted); *LSI Indus., Inc. v. ImagePoint, Inc.*, 279 F. App’x 964, 972 (Fed. Cir. 2008) (rejecting the district court’s construction of “display device” as necessarily including the superfluous limitation of “internal illumination” because other claim terms specifically recited an “illuminated display device”); *but cf. ERBE Elektromedizin GmbH v. Canady Tech. LLC*, 629 F.3d 1278, 1286 (Fed. Cir. 2010) (“no canon of [claim] construction is absolute in its application”) (citation omitted).

Patent Owner also argues that requiring a “database server” to be able to read “a set of files” is necessary to clarify that “a set of files” in its proposed definition must be “database files.” PO Resp. 31–32. Patent Owner’s argument is unpersuasive because the same objective can be accomplished without inserting the extraneous term by directly stating a virtual database must be “a set of database files.” In fact, the ’174 patent describes that a virtual database comprises “database files.” *See, e.g., Ex. 1001*, 3:19–22 (“The virtual databases are ‘virtual’ in the sense that the physical implementation of the *database files* is decoupled from the logical use of the *database files* by a database server.”), 41–42 (“A virtual database

may be created on a database server by creating the *database files . . .*”) (emphases added). Patent Owner appears to argue this is insufficient because “[t]o be a database file, a database server must be able to read, write and understand its contents.” PO Resp. 32 (citing Ex. 2030 ¶ 108). We are not persuaded by this argument because it does not comport with the written description of the ’174 patent.

First, the ’174 patent provides a description of a “database” as follows: “[a] database comprises data stored in a computer for use by *computer implemented applications*.” Ex. 1001, 2:63–64 (emphasis added). Furthermore, the ’174 patent announces that the disclosed invention “can be used for *any database*.” *Id.* at 3:15–18. Hence, a database file may be used by an application or a program but not necessarily a “database server,” which is a server-side program in a client-server system or network. *See id.* at 2:64–67, 17:13–16; *see also* Paper 60 (“Tr.”) 184:1–185:12 (Patent Owner acknowledges that a client-server architecture is not necessary to give meaning to “virtual database” and that “a database server” in its proposed construction “is used synonymously . . . with database application.”).

Furthermore, the ’174 patent describes that various programs that are not a database server can read and write to a database or database file and understand its content. For example, the virtual database manager program at the database storage system can create, read, and write to read/write files representing a virtual database. *See* Ex. 1001, 3:64–66; Ex. 2002, 17:57–63, 18:27–31. Other programs of the database storage system, such as the point-in-time copy manager and the storage allocation manager, can create, read,

and write to database files based on database blocks received from the production system. *See* Ex. 1001, 7:18–35, 7:52–8:3.

Therefore, contrary to Patent Owner’s contention, the written description of the ’174 patent shows that a database does not require a database server reading and writing to the database. Patent Owner’s proposed construction would improperly import a particular embodiment from the specification by reading in the operation of a database server to impermissibly narrow the meaning of the term “virtual database.” As discussed above, features of particular embodiments may not be read into the claims “absent clear disclaimer in the specification.” *Am. Acad. of Sci. Tech Ctr.*, 367 F.3d at 1369.

On the other hand, the specification shows that the readable and writable characteristics are inherent attributes of a virtual database. *See, e.g.*, Ex. 1001, 2:59–62 (“the virtual databases can then be individually accessed and modified as desired”); Ex. 2002, 18:27–28 (“FIG. 10 indicates how storage efficient copies are made to create a *read/write file structure representing a VDB.*”), 17:57–63 (“The virtual database manager [] identifies [] the recent most PIT copy associated with time T_j . . . The *read/write file structure* [] is created [] by making storage efficient copies of the database blocks in the identified PIT copy.”), 18:12–15 (“The virtual database manager [] sends . . . handles to the *read/write file structure* to the associated virtual database system 130.”) (emphases added). Thus, a virtual database must comprise readable and writable database files. However, the phrase “database files” need not be further modified to indicate that the files are intended for use by a computer program because the concept is already included in the definition of “database” provided in the specification, as

discussed above. Therefore, the term “virtual database” is properly construed, to have as the base phrase of its definition, “a set of readable and writable database files.”

b. The Physical Implementation of the Database Files is Decoupled from the Logical Use of the Database Files by the Database Server

Patent Owner relies on the following sentence in the '174 patent to support the second proposed modifying phrase—the “decoupled” aspect of its proposed construction: “virtual databases are ‘virtual’ in the sense that the physical implementation of the database files is decoupled from the logical use of the database files by a database server.” PO Resp. 28–29 (quoting Ex. 1001, 3:19–22). Patent Owner asserts this sentence explicitly defines the use of the word “virtual” in the term “virtual database.” *Id.* However, the '174 patent does not describe expressly what it means for “the physical implementation of the database files [to be] decoupled from the logical use of the database files by a database server.” Citing the testimony of Dr. Shenoy, Patent Owner asserts the cited sentence describes one of the key features of the invention of the '174 patent, which is applying the general concept of virtualization—“the *decoupling* of a computing process *from the platform* on which it operates”—to databases. *Id.* (emphases added) (citing Ex. 2030 ¶¶ 23–26, 100). Petitioner asserts Patent Owner confuses “platform virtualization” with “database virtualization” or “virtualizing data.” See Reply 16–17. We agree with Petitioner that “platform virtualization” is unrelated to the concept of “virtual database” as claimed and described in the '174 patent.

In his Declaration, Dr. Shenoy testifies that the Java programming language running on a Java Virtual Machine as described in an Oracle

document explains the general concept of platform virtualization. See Ex. 2030 ¶¶ 24–25 (citing Ex. 2003, 451–452, Fig. 24-7). Patent Owner’s discussion of Java Virtual Machine illustrates the basic problem with Patent Owner’s argument—that it is directed to the features or functions of software, not a database. For example, Dr. Shenoy explains that the “decoupling [of] a computing process from the platform on which it operates” is achieved by the Java Virtual Machine, which “is *software* that emulates the operation and interface of a physical processor.” See Ex. 2030 ¶¶ 23–24 (emphasis added). The ’808 patent also describes that a “virtual machine” is “provided by *platform virtualization software*” or “server virtualization *software*.” Ex. 2002, 7:16–19 (emphases added). Further, the ’174 patent describes a Java Virtual Machine as part of production database system 110 (Ex. 1001, 5:51–53), which is separate from virtual database 220 stored on database storage system 100, as discussed above. Hence, Patent Owner’s discussion of the Java Virtual Machine (PO Resp. 32 n.9) may relate to database servers or other *software* of the disclosed system, but it is *not* directed to a virtual database, which is a set of *files* as expressly recited in the claims and described in the specification (*see, e.g.*, Ex. 2002, 6: 49–51; Fig. 2a) and therefore is not persuasive even as an analogy.

Other examples of “decoupling” Dr. Shenoy discusses similarly relate to the function or operation of various software or programs, not database files. For instance, Dr. Shenoy testifies that, because the system of the ’808 patent operates at the “application level” using APIs, the database storage system of the invention does not depend on and need not concern itself with the underlying details of the physical storage system. Ex. 2030 ¶ 72 (quoting Ex. 1001, 5:19–24). The cited portion of the ’174 patent describes,

however, the APIs are provided by “production system library 285” and “vendor interface module 235,” which are programs or software code modules residing at the production system. *See, e.g.*, Ex. 1001, col. 5:24–27 (“An example of a vendor interface module is the *program code* of a database server provided by vendor ORACLE that implements RMAN APIs.”) (emphasis added); Ex. 2002, 13:10–12 (“the production system library 385 includes *code* to analyze the structures of the files of the database stored in the data store 350”) (emphasis added).

Dr. Shenoy similarly testifies that “each database (130c)” depicted in Figure 1 of the ’174 patent is “virtual” (Ex. 2030 ¶ 101; *see also* PO Resp. 29 (citing Ex. 2030 ¶ 101)) when, in fact, Figure 1 shows the part identified by Patent Owner is Virtual Database *System* 130(c), *not* the virtual database, which is stored in a different system, namely, Database Storage System 100. As described in Figure 2a of the ’808 patent reproduced above, virtual database *system* 130 is separate from and external to *virtual database* 220 stored on database storage system 100. Further, the ’808 patent describes that virtual database *system* 130 includes no database or database files but, rather, comprises a database server and a VDB system library, both of which are software modules or programs. *See* Ex. 2002, 12:3–13 (“A virtual database system 130 includes a database server 360 and a VDB system library 380 . . . The VDB system library 380 contains *program code* for processing requests sent by the database storage system 100.”) (emphasis added), Fig. 3. Hence, Patent Owner appears to conflate the function of the *software* of a virtual database *system* with the meaning of “a virtual database,” which is a set of *files* that resides in a database storage system that is separate from the virtual database system.

Petitioner argues that Patent Owner’s platform virtualization argument is misguided because it conflates the meaning of a “virtual database” with the disclosed embodiments of production database *system* and the virtual database *system*. Reply 16–17. We agree with Petitioner, but our analysis underscores a related salient point, which is that Patent Owner’s arguments are directed to the software part of the disclosed system that is separate from and external to a virtual database. As described in the ’174 patent, a database system generally comprises a database, which is data stored in storage, and database software, such as a database servers or other program that accesses the database. See Ex. 1001, 2:63–67; 5:7–36. Dr. Shenoy similarly describes a database system as comprising a database and database software.

A database system is typically a computer that includes the stored data itself, i.e. the database, as well as database management software, often shortened to DBMS. A DBMS includes software for accessing the data in the database and modifying that data to add, remove or change stored data. A DBMS may run as or include an autonomous program which handles client requests to the database and which is often called a database server.

Ex. 2030 ¶ 34 (emphases added). Hence, the platform virtualization or the storage virtualization or abstraction argued by Patent Owner as demonstrating what is meant by the phrase “the physical implementation of the database files is decoupled from the logical use of the database files by a database server” is a function or feature of the *software part* of a database system—be it Java Virtual Machine, a database server, or various software modules providing APIs—which is separate from and external to a database, i.e., the *data part* of a database system. Therefore, contrary to Patent

Owner's argument, virtualization of a computing platform or storage by software is unrelated to and cannot be equated to virtualization of data, i.e., the subject matter claimed by the term "virtual database."

In effect, Patent Owner's platform virtualization argument calls for improperly importing the function or operation of the software of the disclosed system into the meaning of a virtual database, which is a set of files separate from and external to the software part. If patentee intended to claim the function or operation of the software of the disclosed system, it could have done so by explicitly claiming the subject matter. Not having done so, Patent Owner may not import the array of functions and intelligence embodied in the described software into a single term "virtual database" to impart a very different meaning to the claims from what is indicated by the plain language of the claims and the written description of the patent. Patent Owner asserts that, by discounting Patent Owner's platform virtualization argument, the preliminary construction reads out the "virtual" requirement from the term "virtual database." PO Resp. 31 (citing Inst. Dec. 20). Patent Owner's argument is unpersuasive because our refusal to read the functions and features of the software modules into the meaning of "virtual database" does not equate reading out the word "virtual" from the term. On the contrary, as discussed below, the preliminary construction is consistent with the language of the claims and the written description of the '174 patent.

Turning to the claim language and the written description, the challenged independent claims recite, with emphasis added, creating "a . . . set of files for the . . . virtual database, each file in the . . . set of files [is]

linked with . . . database blocks.” Similarly, the Specification describes as follows:

A virtual database may be created on a database server by creating the database files for the production database corresponding to the state of the production database at a previous point in time, as required for the database server. The files corresponding to the virtual database are made available to the database server using a file sharing mechanism, which links the virtual database to the appropriate database blocks stored on the storage system.

Ex. 1001, 3:41–48 (emphasis added); *see also* Ex. 2002, 2:24–27 (making essentially the same statement). Hence, the database files of a virtual database are linked or mapped to the database blocks associated with the database being virtualized.

Looking to the Specification for further clarification, Figure 10 of the ’808 patent depicts “how . . . to create a read/write file structure representing a VDB.” Ex. 2002, 18:27–28. When the VDB file structures are created, “the blocks V11, V12, . . . , V25 may be implemented as *pointers* to the actual database block that stores the data.” *Id.* at 18:47–49 (emphasis added). The ’808 patent further describes:

For example, V11 represents the information in block F1 and since the block F1 was never updated during copies made at time T1 and T2, *V11 points at F11*. V12 represents the information in block F2 and since F2 was updated at time T1, *V12 points at the block F22*. Similarly, *V13 corresponds to block F3* that was updated at time T2 and *points at the block F33*.

Id. at 18:49–55 (emphases added). Hence, blocks V11, V12, etc. of a virtual database do not contain ordinary data, but, instead, point to the location or address of the actual database blocks stored in a physical storage device. Another example implies that writing to a pointing block in a database file,

such as V11, which is stored at one location, actually results in writing to a copy of the original block of data F11 at another physical location:

For example, if the virtual database system 130 writes to the block V11, space is allocated and block F11 copied to the allocated block. Hence, the original copy of the block F11 is maintained as a read only copy and the virtual database system 130 is allowed to write to a copy of the appropriate database block created specifically for the virtual database system 130.

Id. at 19:51–57. The '808 patent further describes “[s]ince the [virtual database file] structure 1050 illustrated in FIG. 10, structure 1150 illustrated in FIG. 11, or structure 1350 illustrated in FIG. 13 are read/write structures, the virtual database system 130 is allowed to read from these structures as well as write to them.” *Id.* at 19:44–48. In addition, “[a] database block that is not written to by virtual database systems 130 may be shared by several virtual database systems without being copied for a specific virtual database systems 130.” Ex. 1001, 14:50–53.

Thus, the '808 patent indicates that the database files are “virtual” in the sense that they create the illusion of allowing data to be written to pointing blocks in the virtual database files, but the system actually writes the data elsewhere to another physical location specified by the pointing blocks such as V11 and V12. Ex. 2002, 19:44–53, Figs. 10–12.

Figure 9 also shows how database blocks may be shared by file structures created for several virtual databases. Ex. 1001, 14:1–3. File Structures 950(a) and 950(b) include blocks V11–V14 and U11–U14 respectively. *Id.* at 14:3–8. Each of these blocks is implemented as a pointer to the actual database block storing the relevant data. *Id.* Thus, several blocks in the virtual database may share the same database blocks—

V11 and U11 both point to the same copy of F11. Thus, the '174 patent indicates that the database files are “virtual” in the sense that they create the illusion of storing data, but the system actually stores the data elsewhere in another physical location (F11) specified by the pointing blocks such as V11 and U11. The '174 patent also describes the procedure for writing to a shared database block, such as V11: “For example, if the virtual database system 130 writes to the block V11, space is allocated and block F11 copied to the allocated block.” *Id.* at 14:37–39.

Based on the disclosure in the '174 patent discussed above, “the logical use of database files” can be understood as reading from or writing to database blocks by reading from or writing to virtual database file structures that point to the actual database blocks. Hence, as described in the '174 patent, decoupling of physical implementation of the database files, i.e., actual database blocks stored on a physical storage device, from the logical use of the database files, i.e., accessing the database blocks through virtual database files, is accomplished by using pointers to map blocks in virtual database files to physical addresses for database blocks stored in physical storage devices.

In fact, although not discussed in the Patent Owner Response, Dr. Shenoy acknowledges that Figure 10 illustrates the “decoupling of the physical storage of the database from its logical use” in part, because “common database blocks are shared between different point-in-time copies. This is imperceptible to the . . . user.” Ex. 2030 ¶ 67. Similarly, Dr. Shenoy testifies that “the physical storage for the virtual database is decoupled from the logical view of the database . . . by *mapping* blocks captured at various points in time to [virtual database files]” (*id.* ¶ 106) (emphasis added).

Nonetheless, Patent Owner argues that the preliminary construction is erroneous because database files of any database, including a source database, can be “mapped to physical addresses for stored database blocks.” PO Resp. 31 (citing Ex. 2030 ¶ 108–109). Apparent inconsistencies in Dr. Shenoy’s testimony aside, Patent Owner’s argument is unpersuasive because the ’174 patent expressly claims and describes that a source database can be a virtual database. See Ex. 1001, 14:20–21; 15:20–22; claim 13. To the extent that Patent Owner is arguing the preliminary construction does not distinguish a non-virtual source database from a virtual database, the concern can be addressed by expressly indicating that the stored database blocks are associated with another database since an ordinary, non-virtual database consists of its own database blocks and would *not* be created by mapping or pointing to stored blocks of a different database. This tracks the claim language of the challenged claims, which define how to create a VDB, which in turn, tracks Dr. Shenoy’s testimony. See Ex. 2030 ¶¶ 63, 106.

Furthermore, although not argued by the parties, it is not necessary to require that the mapped database blocks are associated with a point-in-time copy of a database because the challenged independent claims expressly recite virtual database files wherein “at least some of the database blocks are associated with multiple point-in-time copies of the source database.” See *Digital-Vending Servs. Int’l*, 672 F.3d at 1274–75. Therefore, based on the foregoing discussion, the term “virtual database” is construed as “a set of readable and writable database files capable of being mapped to physical addresses for stored database blocks associated with another database.”

C. Obviousness Grounds

Petitioner asserts that the challenged independent claims—1, 27, and 28—are obvious over a combination of Edwards, Edwards II, Neto, and Klivansky (collectively, the “NetApp References”) and the challenged dependent claims—4, 5, 29, and 34—are obvious over the NetApp References combined with Hart.

1. Overview of the NetApp References

The NetApp References relate to a system created by Network Appliance, Inc. (“NetApp”). *See generally*, Exs. 1003 (“Edwards”), 1004 (“Neto”), 1005 (“Edwards II”), 1006 (“Klivansky”). Edwards II is a patent assigned to Network Appliance, Inc. and each of Edwards, Neto, and Klivansky are articles that refer to NetApp, NetApp, Inc., or Network Appliance, Inc. on their cover page. *See* Ex. 1003; 1004; 1006. NetApp’s products include the Data ONTAP operating system, which includes a proprietary file system called Write Anywhere File Layout (“WAFL”). Ex. 1003, 10; Ex. 1004, 1, 9; Ex. 1005, 2:19–22; Ex. 1006, 3–4; PO Resp. 35 (citing Ex. 2030 ¶ 116).¹⁷ All of the NetApp References discuss using the WAFL file system to create copies of instances of file systems, otherwise known as volumes. *See, e.g.*, Ex. 1003, 10 (“A volume is an instantiation of the file system. . . . The WAFL file system implements these volumes”); Ex. 1004, 9, 13; Ex. 1005, 3:31–35; Ex. 1006, 4–6.

Patent Owner argues that Petitioner has not properly shown that the NetApp References describe “exactly the same NetApp” technology because

¹⁷ When referencing Edwards, Exhibit 1003, we refer to the pagination inserted by Petitioner in the bottom, right-hand corner of each page.

the four documents “do not reference each other,” “describe different embodiments of technology,” and “bear dates years apart.” PO Resp. 4–7. According to Patent Owner, the common terms used throughout the documents, such as FlexVol and FlexClone are NetApp trademarks associated with products and technologies that have changed over time. *Id.* at 5. Specifically, Patent Owner asserts that Edwards and Neto are dated several years after Klivansky and describe “new features” establishing that they do not describe the same embodiment addressed in the older documents. *Id.* at 5–6.

We are persuaded that a preponderance of evidence demonstrates that the four NetApp References are sufficiently related for purposes of this obviousness analysis. Patent Owner does not dispute that all four documents refer to NetApp products and technologies that include FlexVol, FlexClone, Snapshots, and the ONTAP operating system. *See* Ex. 1018 ¶¶ 35–43; PO Resp. 35–39. Even if the documents refer to various versions of those products and technologies, we are persuaded that a person of ordinary skill in the art would have been motivated to look to the teachings of the four references based on express disclosures in the references themselves.

For example, the basics of NetApp storage and cloning are described by Edwards and Edwards II. Pet. 30–31 (citing Ex. 1018 ¶¶ 102–103). Although the patent discussing this cloning technology (Ex. 1005, filed April 30, 2004) was filed prior to the publication of an article (Ex. 1003, dated June 22–27, 2008) on the same topic, we are not persuaded that this time difference would deter a person of ordinary skill from combining the teachings of the two references. This is true even if the Edwards article describes updates to the cloning technology not discussed by Edwards II.

And Edwards explicitly discusses the use of writable Snapshot copies in database environments to make “copies of a production database for development or test purposes.” Ex. 1003, 14. A person of ordinary skill in the art looking to implement such a solution would have looked to other NetApp publications such as Neto, which explicitly states that it “is intended for use by ‘Oracle database administrators, storage administrators, and architects implementing a backup, recovery, and cloning solution for Oracle databases running on NetApp storage’” (Pet. 30 (quoting Ex. 1004, 4)) and Klivansky, which describes its contents as being useful as “a way of understanding the full landscape of FlexClone benefits, as a refresher of specific FlexClone operations, or as a guided tutorial that filer administrators can try” (Pet. 31 (quoting Ex. 1006, 25)). Given these explicit statements, we are persuaded that a person of ordinary skill would look to combine the teachings of Klivansky with the teachings of the other references even if Klivansky, published in 2004 (Ex. 1006), discussed an older version of the cloning technology.

a. Overview of Edwards

Edwards describes “virtualization,” “a well-known method of abstracting physical resources and of separating the manipulation and use of logical resources from their underlying implementation.” Ex. 1003, Abs. Specifically, Edwards discusses “virtualiz[ing] file volumes in the WAFL file system” to create FlexVol volumes. *Id.* These FlexVol volumes essentially comprise a file system that points to underlying data storage on storage disks:

[A] FlexVol volume is a file system created *within* a file on an underlying file system. A hidden file system spans a pool of storage, and we create externally visible volumes inside files on

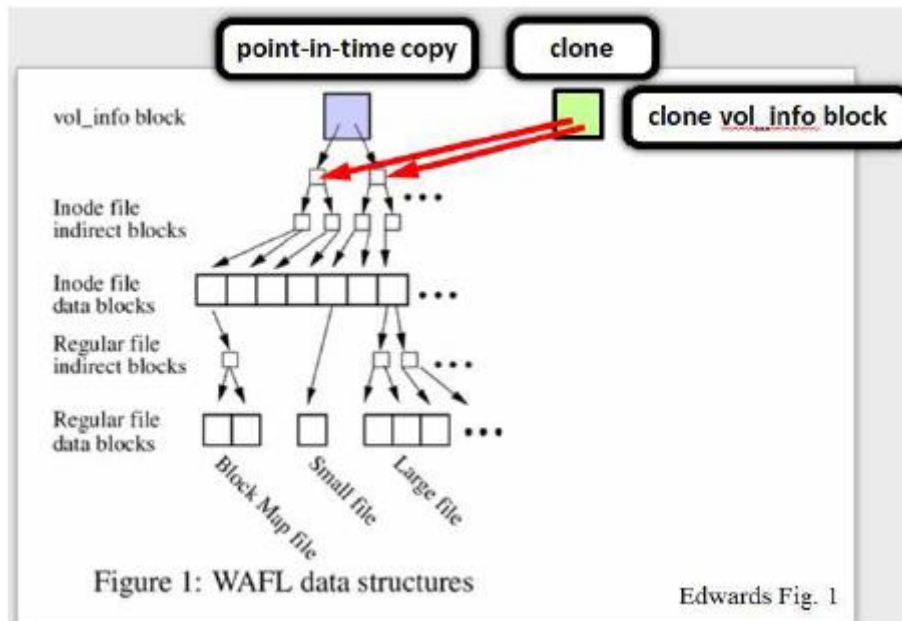
this file system. This introduces a level of indirection, or *virtualization between the logical storage space used by a volume and the physical storage space* provided by the RAID subsystem.

Ex. 1003, 11 (second emphasis added). The FlexVol volumes “are managed independent of lower storage layers.” *Id.* Abs.

Edwards further explains this virtualization technology is used “to implement writable Snapshot copies (called [clones or] *FlexClone volumes*).” *Id.* at 10. A clone volume “inherits pointers to the complete file system image stored in the original Snapshot copy” of an original FlexVol volume. *Id.* at 15. The Snapshot copies are point-in-time copies: “The only differences between a Snapshot copy and the live file system are the blocks that have been modified since the Snapshot copy was created (and the metadata that points to them).” *Id.* at 11. “WAFL Snapshot copies provide consistent point-in-time copies of a volume.” *Id.* at 14. Although a Snapshot copy is read-only, combining Snapshot and FlexClone technologies provides writable Snapshot copies since “[i]n database environments . . . it is often desirable to make writable copies of a production database for development or test purposes.” *Id.*

Edwards explains that “[c]reating a clone volume is a simple process.” Ex. 1003, 15. A container file for the new clone volume (or FlexClone volume) is created and seeded “with a vol_info block that is a copy of the vol_info block of the snapshot copy on which the clone is based.” *Id.* Because vol_info block is the root of the “tree of blocks that form the Snapshot copy, the clone inherits pointers to the complete file system image stored in the original snapshot copy.” *Id.* Petitioner provides an annotated

version of Edwards's Figure 1, to demonstrate this step in cloning, as follows:



Pet. 16. Figure 1, as annotated by Petitioner, depicts WAFL data structures of a Snapshot copy and a clone volume based on the Snapshot copy. The new vol_info block, “contains the inode describing the inode file, which in turn contains the inodes for all of the other files in the file system, including the other metadata files.” Ex. 1003, 10. Edwards further discloses that multiple clones of a volume may be created: “[v]olume clones also present a natural case for thin provisioning, since users will often want many clones of a given volume.” *Id.* at 15.

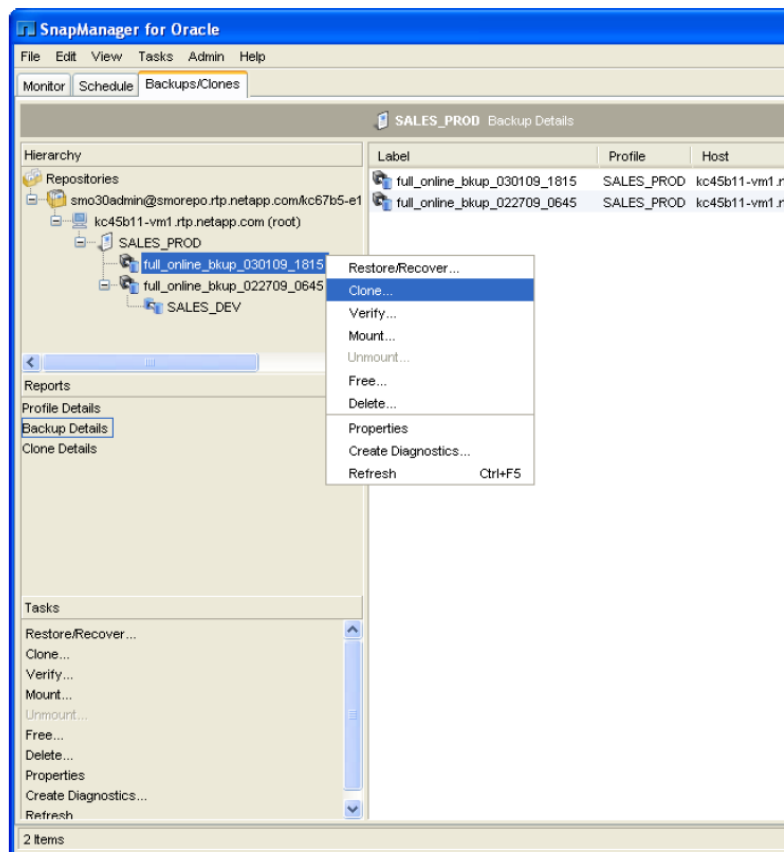
b. Overview of Edwards II

Like Edwards, Edwards II discusses the WAFL file system and describes creating writable copies of virtual volumes by using snapshots of FlexVol volumes. Ex. 1005, Abs., 3:31–57, 6:49–53. It specifically discusses that multiple snapshot copies of a database may be created on a

regular schedule that share the same file system blocks. *Id.* at 3:44–49, 4:26–38. Extra storage blocks are required only for any differences between the copies. *Id.* at 4:38–39.

c. Overview of Neto

Neto describes a NetApp product called SnapManager for Oracle. Ex. 1004, 1. SnapManager uses other NetApp products, including Snapshot and FlexClone, to provide “quick, space-efficient cloning of Oracle databases.” *Id.* SnapManager has a GUI that allows a user to create these copies. *Id.* at 42. Neto shows an example of an administrator creating two virtual databases based on two snapshots. *Id.*; Ex. 1018 ¶¶ 72–73. The figure from page 42 of Neto is partially reproduced below.



Case IPR2015-00128

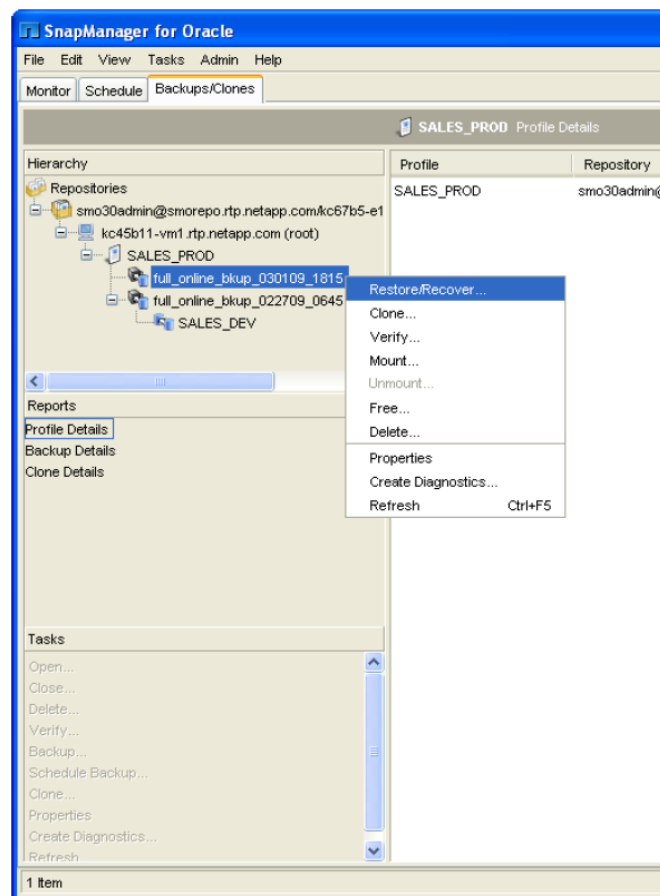
Patent 8,468,174 B1

Ex. 1004, 42. “To clone a database from an existing backup using the SnapManager GUI, right-click on the backup and then select ‘Clone...’.”

Id. After creating a copy, Neto discloses mounting it on a database host. *Id.* at 50.

Neto also describes using SnapManager’s GUI to restore a database to a selected, prior point in time based on a corresponding backup. *Id.* at 35.

The figure from page 36 of Neto is partially reproduced below.



Id. at 36. “To restore and recover from a backup using the SnapManager GUI, right-click on the backup and select ‘Restore/Recover...’.” *Id.*

d. Overview of Klivansky

Klivansky is “a thorough introduction to FlexClone volumes.” Ex. 1006, Title. In one of several hypothetical scenarios, Klivansky describes creating multiple FlexClone volumes of a source volume and mounting each volume to different servers. Ex. 1006, 18–19. In another example, Klivansky maps a clone volume to a testing server “via iSCSI.” *Id.* at 25.

2. Overview of Hart

Hart is a patent assigned to Apple relating to a “user interface for electronic backup.” Ex. 1007, Title. According to Petitioner, the subject matter of Hart has been commercialized as a backup feature in Apple’s Mac computers. Pet. 26 (citing Ex. 1018 ¶ 96). Hart discloses allowing a user to select one of any stored backups of their computer files in order to restore their computer to a particular time. Ex. 1007, ¶¶ 30, 50. Hart discloses an implementation of the GUI for this process uses a timeline—“a vertical column of tick marks.” *Id.* at ¶¶ 8, 81. Figure 16 of Hart is reproduced below.

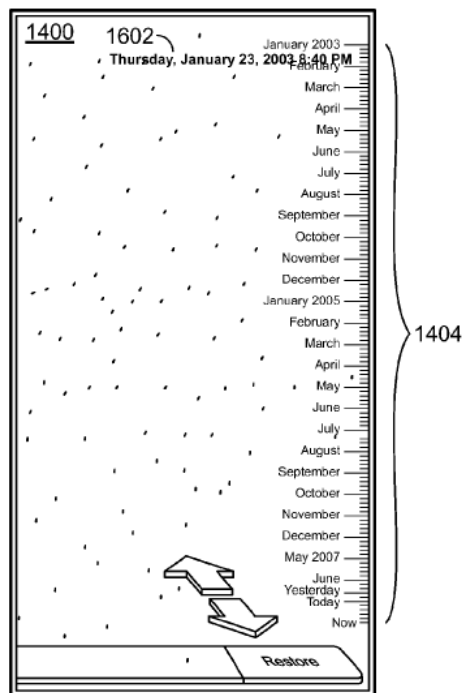


FIG. 16

Figure 16 “shows an implementation of timeline 1402 where the cursor has moved to particular tick mark near the top”—between January and February 2003.” *Id.* at ¶ 90.

3. Claims 1, 27, and 28

Petitioner asserts that independent claims 1, 27, and 28 are unpatentable as obvious over all four NetApp References combined. Pet. 37–57; ’136 Pet. 54–58. Claim 27 differs from claim 1 only slightly in that it claims a “computer-implemented system for creating a virtual database” with several “manager modules” that are configured to perform the steps recited by the method of claim 1. Similarly, claim 28 differs from claim 1 only in that it claims a “computer program product” with several “manager modules” that are configured to perform the steps recited by the method of

claim 1. Except where specifically noted, we do not differentiate between the independent claims in our discussion below.

- a. *“receiv[ing] information identifying a source database for providing data for storage in a first virtual database and a second virtual database being created” and “recei[ing] information identifying a first/second point in time”*

Patent Owner argues that Petitioner does not sufficiently show “receiv[ing] information identifying a source database for providing data for storage in a first virtual database and a second virtual database being created” and “receiv[ing] a first/second point in time” (“the receiving limitations”) recited by all the challenged claims.

When discussing the receiving limitations, the Petition discusses both Edwards and Neto. First, the Petitioner discusses Edwards’ disclosure of “creating multiple virtual (clone) databases based on a source ‘parent volume’ containing a database.” Pet. 37–38, 39, 48–49 (citing Ex. 1003, 14–15). Petitioner explains that based on this disclosure, “Edwards discloses creating a first virtual (clone) database and a second virtual (clone) database.” *Id.* (citing Ex. 1018 ¶¶ 141, 148). Petitioner then concedes that Edwards does not explicitly disclose the receiving of information identifying the source database for these two databases, but states that it is inherent—“the only way to clone a source database is to identify it for cloning.” *Id.*

Second, the Petition points to Neto as disclosing the receiving limitations. *Id.* at 37–40, 48–49 (citing Ex. 1018 ¶ 142, 149, 174). Specifically, Petitioner points to Neto’s disclosure of the SnapManager GUI receiving information identifying a source database when the user selects the source database in the GUI, and then selecting to clone that database. *Id.*

(citing Ex. 1018 ¶¶ 143–144, 149–150, 174–175). Petitioner points to Neto’s statement that “each developer or QA engineer can be provided with his or her own personal copy of the database” as evidence that Neto discloses that multiple clones of a source database can be created and notes that these copies may be made at different points in time. *Id.* at 39 (quoting Ex. 1004, 42) (citing Ex. 1004, 27; Ex. 1018 ¶ 145). Petitioner then argues, and provides testimony from Dr. Zadok, that a person of ordinary skill in the art would have combined these teachings of Neto with those of Edwards because Neto provides “a step-by-step user guide to implement the same FlexClone technology-based cloning technique.” Pet. 39 (citing Ex. 1018 ¶ 146).

Patent Owner asserts that Petitioner does not identify anything in the prior art references that it relies on for disclosing “for providing data for storage in a first virtual database and a second virtual database being created.” PO Resp. 40 (citing Pet. 37–39; ’136 Pet. 37–39). Patent Owner argues that the disclosure of Neto described above is not sufficient because the “backup snapshot” selected by the user as a source for cloning is “already stored; it is not selected by a user to provide data for storage in a first virtual database and a second virtual database being created, as required by the claims.” PO Resp. 40 (citing Ex. 2030 ¶¶ 139–142¹⁸). Patent Owner asserts that “the three independent claims require identification of a source database *for providing data for storage* in a first virtual database and a second virtual database and *a storage system* which stores the data provided by the source database.” *Id.* at 40–41 (citing Ex. 2030 ¶¶ 139–142). In

¹⁸ The relevant testimony appears to be located at Ex. 2030 ¶¶ 142–143.

other words, Patent Owner argues that this claim limitation excludes situations in which the source database is already stored on the same database storage system on which the two point-in-time virtual databases are going to reside. Instead, according to Patent Owner, this limitation only encompasses situations where the chosen source database is stored somewhere else and, thus, the database blocks from the source database must be received for the point-in-time copies.

Patent Owner also argues that Petitioner maps all three of the receiving limitations to a single action—a user selecting a backup of a database to be cloned on a GUI—but that Petitioner does not point to any disclosure that “explains what information, identifiers or variables SnapManager in fact receives when the user selects a backup on the [GUI].” PO Resp. 41. Thus, according to Patent Owner, Petitioner does not identify of identification of “a source database” or “a point in time” as required by the receiving limitations. *Id.* at 41–42 (citing Ex. 2030 ¶¶ 140–41).

We find a preponderance of the evidence demonstrates that a person of ordinary skill would have found the receiving limitations obvious. We are not persuaded that, as Patent Owner argues, these limitations require “identification of a source database *for providing data for storage*” in the two point-in-time virtual databases or “*a storage system which stores the data provided by the source database.*” PO Resp. 40–41. The plain language of the claims requires only the receipt of “information identifying a source database for providing data for storage [the two virtual databases] being created.” Nothing in these words demands that the data in the source database be stored in any particular location. Nor does it call for moving or receiving data from one storage system to another. This reading of the

receiving limitations is also consistent with the Specification, which states that the “source database presented to the user for selection” may include databases “for which the database storage system 100 *has stored* point-in-time copies and transaction logs” indicating that the data in some source database already resides on the same system as the virtual databases to be created. Ex. 1001, 10:1–5; Reply 9–10, 19 (citing Ex. 1064 ¶ 49); *see also* Ex. 1001, 11:35–39 (“In response to the user selecting a particular source database 410, the user interface provides . . . information describing the various point-in-time copies 430 stored on the database storage system 100.”), Figs. 3, 4. Moreover, claim 13 explicitly recites “[t]he method of claim 1, wherein the source database is a previously created virtual database,” which means claim 1 is at least broad enough to include this circumstance. *See* Reply 10 (citing Ex. 1064 ¶ 10).

We also credit Dr. Zadok’s testimony that a person of ordinary skill in the art would understand Neto as teaching that by selecting an existing backup database in the GUI, both a source database and a point-in-time are simultaneously selected and identified. Ex. 1018 ¶¶ 174–75; Ex. 1064 ¶¶ 51–53. And we agree with Petitioner that the claim does not require the information identifying the source database and the point-in-time to be implemented as separate and distinct variables. *See* Reply 19–20. Thus, we are not persuaded by Patent Owner’s argument that Petitioner does not identify identification of “a source database” or “a point in time” as required by the receiving limitations.

- b. “creating a first/second set of files for the first/second virtual database, each file in the first/second set of files linked with a first/second set of database blocks comprising information stored in*

the source database prior to the first/second point in time”

Each of the challenged claims recited “creating a first set of files for the first virtual database, each file in the first set of files linked with a first set of database blocks comprising information stored in the source database prior to the first point in time” and an identical limitation with all occurrences of the word “first” replaced with the word “second” (“the creating a set of files limitations”). Patent Owner argues that Petitioner does not sufficiently show these limitations disclosed by the prior art references. PO Resp. 42–45; 51–52 (citing Pet. 45; ’136 Pet. 45).

The Petition points to Edwards’s creation of FlexClone volumes as teaching the creating a set of files limitations. Pet. 45–47 (citing Ex. 1003, 9–15; Ex. 1018 ¶¶ 162–63, 166, 168). According to Petitioner, cloning a volume containing a database by creating a new vol_info block “creates a new file system comprising a new set of database files.” Reply 22 (citing Pet. 13–17).

Patent Owner argues that Edwards’s creation of FlexClone volume of a stored snapshot is not “a newly created set of files,” but is “merely a copy of the ‘vol_info’ block of the volume to be cloned with pointers to existing files.” PO Resp. 43. Patent Owner also argues that the inode files, to which Edwards’s new vol_info block points, do not point to stored database blocks, but instead point to inodes, which are not database blocks. *Id.* at 44.

Patent Owner concedes that “[t]o create a set of files means to generate a set of files that did not exist before.” PO Resp. 42 (citing various dictionary definition of “create”). But Patent Owner goes on to construe this to mean that *every file in the set of files* must be new. *Id.* at 43. We read the

plain language of the creating limitation to require simply that a new set of files be created—with no additional requirement that every file within that set be new.

We also find that Edwards discloses the new vol_info block itself is a new file and when included with other inodes, themselves files, this creates a new set of files. *See, e.g.*, Ex. 1003, 15 (“WAFL creates the files required for a new FlexVol volume . . . [and] seeds the container file of the clone with a vol_info block that is a copy of the vol_info block of the Snapshot copy on which the clone is based.”); Pet. 13–17; Reply 22–23; Ex. 1018 ¶¶ 48–64, 161–168, 180; Ex. 1064 ¶¶ 58–62. And, based on our interpretation of the term “database blocks,” we find by a preponderance of evidence that the inodes in the new set of files point to database blocks comprising information stored in the source database prior to the first/second point in time. *See, e.g.*, Ex. 1003, 15 (“[T]he clone inherits pointers to the complete file system image stored in the original Snapshot copy.”); Pet. 13–17; Reply 22–23; Ex. 1018 ¶¶ 48–64, 161–168, 180; Ex. 1064 ¶¶ 58–62.

We conclude that a preponderance of the evidence demonstrates that a person of ordinary skill would have found the creating a set of files limitations obvious.

c. “database blocks”

Patent Owner argues that the proposed combination of references does not render obvious the “database blocks” limitations even under Petitioner’s proposed construction, which, as discussed above, we adopt. PO Resp. 45–48. Specifically, Patent Owner argues that Edwards’s WAFL data blocks are not equivalent to database blocks because they are not “used by a database.” Instead, Patent Owner asserts that no database has knowledge of

the WAFL data blocks “existence, where they are located, or how to interact with them.” PO Resp. 46 (citing Ex. 2030 ¶ 162).

Patent Owner’s argument, however, is premised on the previously discussed “application level” versus “storage level” distinction—the assumption that “[d]atabase servers only use database blocks which are data structures that can be accessed or used at the application level” and “only the database server,” but not the WAFL file system, “has knowledge of and manipulates the database block as an independent unit of data.” *Id.* at 46–47 (citing Ex. 2030 ¶¶ 153–55). As discussed above, we reject this premise as divorced from the written description of the ’174 patent.

We conclude that a preponderance of the evidence demonstrates that a person of ordinary skill would have found the claimed database blocks obvious over WAFL data blocks disclosed by the NetApp References.

d. “receiv[ing] information identifying a first/second destination database server” and “mount[ing] the first/second set of files on the first/second destination database server”

Patent Owner argues that the proposed combination of references does not render obvious “receiv[ing] information identifying a first/second destination database server for accessing the first/second database being created” and “mount[ing] the first/second set of files on the first/second destination database server” (“the mounting limitations”). PO Resp. 51–54.

When discussing the mounting limitations, the Petition discusses both Edwards and Klivansky. First, Petitioner concedes that Edwards does not explicitly disclose making a virtual database accessible to a destination database server, but “strongly suggests this mounting step” to a person of ordinary skill “because a NetApp storage appliance’s *main* purpose is to be

accessible as a file/storage server using file-server or storage protocols to other computers, including database servers.” Pet. 40–41 (citing Ex. 1018 ¶ 151).

Second, the Petition points to Klivansky as disclosing the mounting limitations. *Id.* at 41–42, (citing Ex. 1018 ¶¶ 152, 153). Specifically, Petitioner points to Klivansky’s disclosure of examples in which users input commands containing identifying information of a destination database server for accessing the virtual database being created. *Id.* at 41. Petitioner specifically describes one example (“the Dave’s example”), in which a cloned Oracle database is stored in a storage server and is mounted to a database server for testing purposes. *Id.* at 41–42 (citing Ex. 1006, 21–25; Ex. 1018 ¶ 152). The command “lun map /vol/NEWfiles/lun0 Testing 10” makes a clone volume named “NEWfiles” accessible to a testing server named Testing 10, which has on it a cloned volume containing a database. *Id.* at 42 (citing Ex. 1018 ¶ 153). Later, Petitioner discusses another example (“the Alice example”), in which a clone of an Oracle database is made accessible to a testing server via SCSI. Pet. 47–48 (citing Ex. 1006, 18; Ex. 1018 ¶¶ 170–71). Given these disclosures, Petitioner concludes, and provides testimony from Dr. Zadok, that a person of ordinary skill “using Edwards to create a virtual database system would have looked to Klivansky because it is a NetApp user guide that provides step-by-step instructions to clone and mount virtual databases for test and development purposes, and mounting a clone was an important purpose of Edwards’s cloning technology.” Pet. 42, 48, 50–51, 54; Ex. 1018 ¶ 172; *see also* Ex. 1018 ¶ 154:

A POSITA would have combined the disclosure of Edwards with Klivansky because the fundamental purpose of creating a clone database is to make it accessible to a database server and Klivansky is a NetApp publication that teaches how to mount the clone databases described in Edwards to a database server.

Patent Owner argues that the Dave and Alice examples are different embodiments and that the Alice example “has nothing to do with the Dave example,” “does not involve data from an Oracle database and it does not mention, use or refer to the testing server Testing 10, which Petitioner maps to the first destination server.” PO Resp. 52. Thus, Patent Owner concludes that Petitioner improperly combines two separate and distinct embodiments “to establish anticipation.” *Id.* at 53–54 (citing Ex. 2030 ¶¶ 149–50).

We find a preponderance of the evidence demonstrates that a person of ordinary skill would have found the mounting limitations obvious. We are not persuaded by Patent Owner’s arguments to the contrary for several reasons. First, Patent Owner attempts to hold Petitioner to a showing of anticipation for the mounting limitations because the Petitioner argues that “Klivansky *discloses*” these limitations. *Id.* Patent Owner, however, provides no authority to support this theory. In fact, when read in context, this portion of the Petition and the testimony proffered in support clearly propose that a person of ordinary skill would find the mounting limitations obvious in view of the teachings of Edwards and Klivansky combined. Second, we find that the Dave and Alice examples are not separate embodiments, but instead are two examples of how to use FlexClone technology. Ex. 1006, 25; Reply 21–22.

We conclude that a preponderance of the evidence demonstrates that a person of ordinary skill would have found the claimed mounting limitations obvious over the combination of Edwards and Klivansky.

e. “virtual database”

Patent Owner argues that the proposed combination of references does not render obvious “creating the first/second virtual database” even under Petitioner’s proposed construction, which, as discussed above, we adopt. PO Resp. 54–57. Specifically, Patent Owner argues that Edwards’s FlexClone volumes are not equivalent to virtual databases because they are not “a set of database files.” Patent Owner asserts that inodes are not database files because “they are not capable of being mapped to physical addresses for stored database blocks,” but instead are “collections of inodes which may point to file system data blocks.” PO Resp. 56 (citing Ex. 2030 ¶ 170).

Patent Owner’s argument, however, is premised on the previously discussed definition of “database block.” As discussed above, we reject Patent Owner’s proposed construction.

We conclude that a preponderance of the evidence demonstrates that a person of ordinary skill would have found the claimed “creating the first/second virtual database” obvious over FlexClone volumes disclosed by the NetApp References.

f. “module”

Challenged claims 27 and 28 require several “managers” and “manager modules,” including “a user interface manager,” “a storage allocation manager module,” “a virtual database manager module,” and “a file share manager module” (“the module limitations”). Patent Owner

argues that Petitioner does not sufficiently identify these limitations in the prior art references. PO Resp. 59–60 (citing Pet. 56; ’136 Pet. 55–56).

The Petition asserts that “[t]he computer program modules recited in [claims 27 and 28] represent software code embodied on a machine-readable medium.” Pet. 56 (citing Ex. 1001, 18:22–24 (“The described operations and their associated modules may be embodied in software, firmware, hardware or any combinations thereof.”)). Petitioner then explains that “NetApp’s Data ONTAP system comprising of its snapshot, FlexVol, and FlexClone technologies described in Edwards, Edwards II, Neto, and Klivansky are computer features implemented in code loaded onto a physical medium programmed to perform each of the stated functions of Claim 1.” *Id.* (citing Ex. 1018 ¶ 193).

According to Patent Owner, it is not enough for Petitioner to rely on the fact that the NetApp references describe “computer features implemented in code.” PO Resp. 59. Instead, “Petitioner must show where each and every module is disclosed in the documents” in order to show that the claimed functions are performed by “separate modules.” *Id.* (citing Ex. 2030 ¶ 181). We do not agree with Patent Owner’s narrow reading of the module limitations. The ’174 patent explicitly notes that the use of these terms is not meant to be restrictive, stating:

Some portions of this description describe the embodiments of the invention in terms of algorithms and symbolic representations of operations on information. These algorithmic descriptions and representations are commonly used by those skilled in the data processing arts to convey the substance of their work effectively to others skilled in the art. These operations, while described functionally, computationally, or logically, are understood to be implemented by computer programs or equivalent electrical circuits, microcode, or the like.

Furthermore, it has also proven convenient at times, to refer to these arrangements of operations as modules, without loss of generality. The described operations and their associated modules may be embodied in software, firmware, hardware, or any combinations thereof.

Any of the steps, operations, or processes described herein may be performed or implemented with one or more hardware or software modules, alone or in combination with other devices.

Ex. 1001, 18:11–28 (emphasis added). Based on this disclosure and the language of the claims, we find that the terms “manager” and “manager module” in claims 27 and 28 do not require separate or exclusive modules for each articulated function.

Patent Owner also argues that Petitioner does not identify any module in the prior art references that corresponds specifically to “a storage allocation manager module” because this module “stores database blocks retrieved from a production database system.” PO Resp. 60 (citing Ex. 2030 ¶ 181). We are unclear, however, where the requirement of “production database” in Patent Owner’s argument comes from. Patent Owner does not explain it further, and it is not used in the claim language, which refers only to a “source database” and not a “production database.” To the extent this is an extension of Patent Owner’s argument that the claims exclude situations in which the source database is an already created virtual database, we are unpersuaded for the reasons discussed above.

We conclude that a preponderance of the evidence demonstrates that a person of ordinary skill would have found the module limitations obvious over the combination of the NetApp references and Klivansky.

g. Conclusion

In the Decision to Institute, we determined that Petitioner had shown a reasonable likelihood of prevailing on this proposed ground of unpatentability. Inst. Dec. 25–28. We have reviewed both parties’ arguments and supporting evidence, including the disclosure of all four references and the testimony of both Dr. Zadok and Dr. Shenoy. Despite the counter-arguments in Patent Owner’s Response, and the evidence cited therein, for the reasons discussed above, Petitioner has shown, by a preponderance of the evidence, that each of claims 1, 27, and 28 is unpatentable under 35 U.S.C. § 103 as directed to subject matter obvious over the combination of Edwards, Edwards II, Neto, and Klivansky.

4. Claims 4, 5, 29, and 34

Claim 4 depends from claim 1, claim 29 depends from claim 27, and claim 34 depends from 28 and further recites “presenting a geometric shape, wherein a position in the geometric shape corresponds to a time value; receiving a selection of a position in the geometric shape; and mapping the selection of the position in the geometric shape to the point in time selected.” Claim 5 depends from claim 4 and further adds “wherein the geometric shape is a line.” Petitioner contends that these claims are rendered obvious by the combination of Edwards, Edwards II, Neto, Klivansky, and Hart. Pet. 57–60 (citing Ex. 1018 ¶¶ 190–199); ’136 Pet. 59–60 (citing Ex. 1018 ¶¶ 197–199). Patent Owner does not separately argue these claims, but relies on the same arguments as those advanced with respect to the independent claims. PO Resp. 40–60.

In the Decision to Institute, we determined that Petitioner had shown a reasonable likelihood of prevailing on this proposed ground of

unpatentability. Inst. Dec. 28–30. We now find that a preponderance of evidence demonstrates that the subject matter of dependent claims 4, 5, 29, and 34 would have been obvious to a person of ordinary skill notwithstanding the additional limitations recited by those claims. In particular, Hart discloses a geometric shape—a timeline in the form of a line—for selecting a point in time and a corresponding snapshot. Ex. 1007 ¶¶ 8, 79, Figs. 5–11, 13–23; Ex. 1018 ¶¶ 97–100.

The Petition further explains that a person of ordinary skill would have found it obvious to modify NetApp’s existing SnapManager GUI (as disclosed in Neto) for Hart’s timeline GUI because they provide similar functionality of managing multiple point-in-time databases and both allow restoring files to their prior stated based on a selected point-in-time. Pet. 34–35 (citing Ex. 1018 ¶ 117–120). We find a preponderance of the evidence supports this assertion. We find persuasive Dr. Zadok’s testimony that the difference between the two references is that “Hart’s GUI used a timeline structure to display stored snapshots, whereas Neto used a list structure to display backups (snapshots), one of which could be selected for restoration by right-clicking it.” Ex. 1018 ¶ 120. And we agree that “[s]ubstituting Hart’s timeline-based GUI for selecting (typically by clicking a mouse) a point-in-time copy for Neto’s right-click based method of selection was nothing more than a simple substitution of one known method of selection with another.” Pet. at 35; *see also* Ex. 1007 ¶ 25 (“[I]t should be apparent that the disclosed implementations can be incorporated in, or integrated with, any electronic device that has a user interface.”). Moreover, we credit Dr. Zadok’s testimony that “Neto’s examples name the point-in-time copies based on the time they were taken and can list them

alphabetically; thus, they were typically arranged in a timeline” and concludes that “[g]iven the similarities . . . it would have been obvious for a POSITA to substitute Neto’s GUI with Hart’s GUI to perform the snapshot backup, restore, and clone functions described in Neto, Edwards, Edwards II, and Klivansky.” Ex. 1018 ¶ 120. We, therefore, find a preponderance of the evidence demonstrates that a person of ordinary skill would have had reason to combine the teachings of the Edwards, Edwards II, Neto, and Klivansky with Hart.

In sum, as with the independent claims, we have reviewed both parties’ arguments and supporting evidence, including the disclosure of all four references and the testimony of both Dr. Zadok and Dr. Shenoy. Despite the counter-arguments in Patent Owner’s Response, and the evidence cited therein, for the reasons discussed above, Petitioner has shown, by a preponderance of the evidence, that each of claims 4, 5, 29, and 34 is unpatentable under 35 U.S.C. § 103 as directed to subject matter obvious over the combination of Edwards, Edwards II, Neto, Klivansky, and Hart.

PATENT OWNER’S MOTION TO EXCLUDE EVIDENCE

Patent Owner moves to exclude Petitioner’s Exhibits 1004, 1020, 1028, 1032, 1033, 1035, 1036, 1037, 1039–1044, 1046, 1048, 1049, 1051, 1054, 1056, 1057, 1059, 1061, and 1062. Paper 37 (“PO Mot. to Exclude”), 1. In *inter partes* reviews, documents are admitted into evidence subject to an opposing party asserting objections to the evidence and moving to exclude the evidence. 37 C.F.R. § 42.64. As movant, Patent Owner has the burden of showing that an Exhibit is not admissible. 37 C.F.R. § 42.20(c).

Patent Owner contends that Petitioner does not rely on Exhibits 1032, 1033, 1035–1037, 1039, 1041, 1051, 1056, 1057, 1059, and 1061. *Id.* at 1 n.2. Petitioner does not dispute this contention. See Paper 47 (“Opp. Mot. to Exclude”). We, therefore, grant Patent Owner’s motion as it pertains to these Exhibits. Further, Petitioner has moved, unopposed, to expunge Exhibit 1028 (see Paper 35), which motion we hereby grant.

Of the remaining objected-to Exhibits, our Final Written Decision discusses and relies only on Exhibits 1004, 1020, 1048, and 1049. We determine, for reasons set forth above, that Petitioner has demonstrated by a preponderance of the evidence that the challenged claims are unpatentable, without need for Petitioner’s additional arguments or evidence in relation to the remaining additional Exhibits. Accordingly, Patent Owner’s motion to exclude Exhibits 1040, 1042–1044, 1046, 1054, and 1062 is *dismissed* as moot.

According to Patent Owner, “Exhibit 1004 is allegedly a technical report by NetApp which Petitioner relies upon for its prima facie case of obviousness.” Mot. to Exclude, 2. Patent Owner moves to exclude the Exhibit (Neto) as not being authenticated pursuant to Federal Rule of Evidence 901. *Id.*

Petitioner contends testimony by Louis Hernandez is sufficient to authenticate the document pursuant to Rule 901(b)(1). Opp. Mot. to Exclude, 14 (citing Exs. 1020, 1048). Patent Owner, however, argues that Mr. Hernandez lacks personal knowledge to testify regarding this issue rendering his statements hearsay. Mot. to Exclude, 8–12 (referring to Exs. 1020, 1048); Paper 50, 5. We note that Mr. Hernandez’s personal knowledge of Neto comes from his review and recognition of that document

as being published during his tenure as an employee of Midwave, a certified distributor and reseller of NetApp. Ex. 1020 ¶¶ 14–18; Ex. 1048 ¶ 3, 6, 11. This is ample basis for him to testify with personal knowledge of the facts under FRE 602. We are not persuaded otherwise by Patent Owner’s characterization of Mr. Hernandez’s cross-examination testimony. See PO Mot. to Exclude, 9–10 (citing Ex. 2031); Paper 38 (citing Ex. 2031)). As to the hearsay objection, Mr. Hernandez’s testimony consists of statements made by the Declarant while testifying in this proceeding — not “hearsay” (Fed. R. Evid. 801(c)) — but sworn testimony that is subject to cross-examination. Indeed, as noted above, Patent Owner cross-examined Mr. Hernandez with respect to that testimony. We, therefore, deny Patent Owner’s motion to exclude Exhibits 1020 and 1048.

Petitioner also contends Neto is self-authenticating. Opp. Mot. to Exclude, 12–14 (citing Rules 901(b)(4), 902(7)). Specifically, Petitioner refers to the “NetApp trade inscriptions such as unique NetApp logos,” “trademark symbols,” and “NetApp trade copyright notices.” *Id.* at 13. Further, Petitioner points to Neto’s contents, which “include repeated references to NetApp and various NetApp technologies.” *Id.* at 14.

We are persuaded that Neto is authenticated at least under Federal Rules of Evidence 901(b)(1) and 901(b)(4). On this record, we accept Exhibit 1004 for what it purports and is alleged to be: A “Technical Report” describing a NetApp product, published by NetApp, for the purpose of providing detailed instructions to its customers and potential customers. Ex. 1004.

The only reason Patent Owner gives for excluding Exhibit 1049 is that it is “untimely under 37 C.F.R. § 42.23(b) and 37 C.F.R. § 42.123.” PO

Mot. to Exclude, 1 n.2. We find that Exhibit 1049, an article titled “Bigtable: A Distributed Storage System for Structured Data,” was properly submitted by Petitioner as evidence in rebuttal to Patent Owner’s claim construction argument for the term “database block” asserting that it must necessarily include metadata. In particular, Petitioner argues this article is relevant to Patent Owner’s assertion that “it is well known to one skilled in the art that a database block necessarily includes metadata.” See PO Resp. 19–23; Reply 11–15 (citing Ex. 1049).

ORDER

Accordingly, it is

ORDERED that claims 1, 4, 5, 27–29, and 34 of the ’174 patent have been shown to be *unpatentable*;

FURTHER ORDERED that Petitioner’s motion to expunge Exhibit 1027 is *granted*;

FURTHER ORDERED that Patent Owner’s motion to exclude evidence is *granted-in-part, dismissed-in-part and denied-in-part*;

FURTHER ORDERED that Exhibits 1032, 1033, 1035–1037, 1039, 1041, 1051, 1056, 1057, 1059, and 1061 shall be expunged; and

FURTHER ORDERED that, because this is a Final Written Decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

Case IPR2015-00128

Patent 8,468,174 B1

PETITIONER:

Robert Steinberg

Jonathan Link

Giri Pathmanaban

Latham & Watkins LLP

bob.steinberg@lw.com

jonathan.link@lw.com

giri.pathmanaban@lw.com

PATENT OWNER:

J. David Hadden

Saina Shamilov

Fenwick & West LLP

dhadden-ptab@fenwick.com

sshamilov-ptab@fenwick.com