United States Court of Appeals for the Federal Circuit

ENFISH, LLC,
Plaintiff-Appellant

v.

MICROSOFT CORPORATION, FISERV, INC., INTUIT, INC., SAGE SOFTWARE, INC., JACK HENRY & ASSOCIATES, INC.,
Defendants-Appellees

2015-1244

Appeal from the United States District Court for the Central District of California in No. 2:12-cv-07360-MRP-MRW, Senior Judge Mariana R. Pfaelzer.

Decided: May 12, 2016

ORION ARMON, Cooley LLP, Broomfield, CO, argued for plaintiff-appellant. Also represented by JAMES P. BROGAN, JANNA FISCHER.

CHAD S. CAMPBELL, Perkins Coie LLP, Phoenix, AZ, argued for all defendants-appellees. Defendants-appellees Microsoft Corporation, Fiserv, Inc., Intuit, Inc., Jack Henry & Associates, Inc., also represented by DAN L. BAGATELL, THEODORE H. WIMSATT; ELIZABETH M. BANZHOFF, AMANDA D.W. TESSAR, Denver, CO.

WILLIAM J. BROWN, JR., Brown, Wegner & Berliner LLP, Irvine, CA, for defendant-appellee Sage Software, Inc. Also represented by MATTHEW K. WEGNER, YUANJUN LILY LI.

Before MOORE, TARANTO, and HUGHES, *Circuit Judges.*

HUGHES, *Circuit Judge*.

Enfish sued Microsoft for infringement of several patents related to a "self-referential" database. On summary judgment, the district court found all claims invalid as ineligible under § 101, some claims invalid as anticipated under § 102, and one claim not infringed. Enfish appeals. We find that the claims are not directed to an abstract idea, so we reverse the summary judgment based on § 101. We find that the "pivot table" feature of the prior art Excel product does not contain the "self-referential" feature of the claims, so we vacate the summary judgment based on § 102. Lastly, we find no error in the district court's determination on non-infringement, so we affirm the summary judgment of non-infringement. We remand the case for further proceedings.

I

Microsoft develops and sells a variety of software products, including the product ADO.NET. At least through the late 1990s and early 2000s, Enfish developed and sold software products, including a new type of database program.

Enfish received U.S. Patent 6,151,604 and U.S. Patent 6,163,775 in late 2000. Both claim priority to the same application filed in March 1995.

The '604 and '775 patents are directed to an innovative logical model for a computer database. A logical model is a model of data for a computer database explaining how the various elements of information are related to

one another. A logical model generally results in the creation of particular tables of data, but it does not describe how the bits and bytes of those tables are arranged in physical memory devices. Contrary to conventional logical models, the patented logical model includes all data entities in a single table, with column definitions provided by rows in that same table. The patents describe this as the "self-referential" property of the database. '604 patent, col. 2 ll. 44–52.

This self-referential property can be best understood in contrast with the more standard "relational" model. With the relational model, each entity (i.e., each type of thing) that is modeled is provided in a separate table. For instance, a relational model for a corporate file repository might include the following tables:

document table,

person table,

company table.

The document table might contain information about documents stored on the file repository, the person table might contain information about authors of the documents, and the company table might contain information about the companies that employ the persons.

Each table in the relational model contains columns defining that table. In the corporate file repository example, the relational model might have the following tables:[1]

Document Table			
ID	**Title**	**Address**	**Author**

Person Table		
ID	**Label**	**Employed By**

Company Table		
ID	**Label**	**Address**

Using this relational model, if a database were to store information about a document called proj.doc, a person called Scott Wlaschin, and a company called DEXIS, then the result might be:

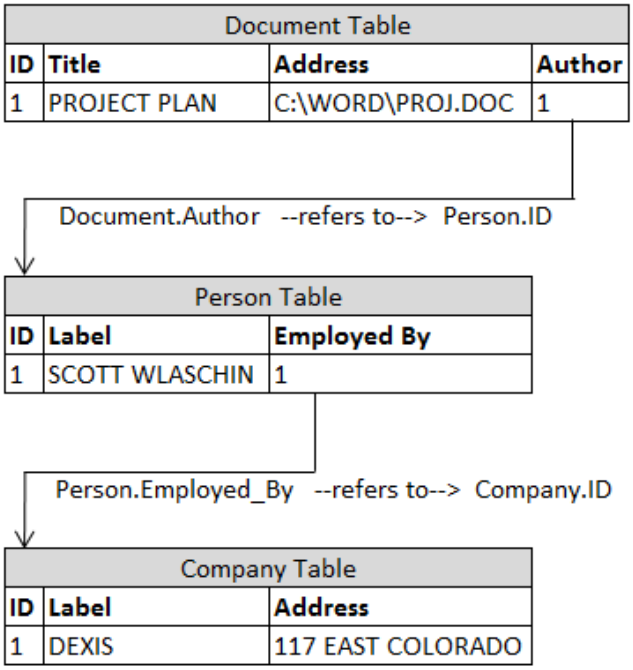
Document Table			
ID	**Title**	**Address**	**Author**
1	PROJECT PLAN	C:\WORD\PROJ.DOC	1

Person Table		
ID	**Label**	**Employed By**
1	SCOTT WLASCHIN	1

Company Table		
ID	**Label**	**Address**
1	DEXIS	117 EAST COLORADO

[1] The figures that follow in this Background section are adaptations of the example tables illustrated in the patents on appeal. *See, e.g.*, '604 patent, Figures 3, 5, 9.

To indicate that Scott Wlaschin is the author of proj.doc and that he is employed by DEXIS, the relational model uses relationships as follows:



Here, the top-most relationship explains that the value for "Author" in the Document table refers to the "ID" column of the Person table. Because the row for proj.doc has AUTHOR = 1, the row in the Person table that has ID = 1 is the author of proj.doc. By this technique, the relational model captures information about each type of entity in a separate table, with relationships between those tables informing the relationships between rows in different tables.

In contrast to the relational model, the patented self-referential model has two features that are not found in the relational model: first the self-referential model can store all entity types in a single table, and second the self-referential model can define the table's columns by rows in that same table. For example, a self-referential model corresponding to the example relational model discussed above might look like the following:[2]

SELF-REFERENTIAL TABLE						
ID	**Type**	**Title**	**Label**	**Address**	**Employed By (#4)**	**Author**
#1	DOCUMENT	PROJECT PLAN		C:\WORD\PROJ.DOC		#2
#2	PERSON		SCOTT WLASCHIN		#3	
#3	COMPANY		DEXIS	117 EAST COLORADO		
#4	FIELD		EMPLOYED BY			

This self-referential table stores the same information that is stored by the example relational model shown above. However, all of the information about documents, persons, and companies are stored in a single table.

Further, an additional row is included in the self-referential table: the row beginning with ID = #4. This row has values of TYPE = "field" and LABEL = "Employed By." Such a row with TYPE = "field" is a special row, because it defines characteristics of a column in that same table. In this case, the row with ID = #4 corresponds to the penultimate column, which is denoted by also marking that column with the ID of #4. The row with ID = #4 defines a single characteristic of the corresponding column, viz., its label. Because the row with ID = #4 has LABEL = "Employed By," we know that the corresponding column is labeled "Employed By," as seen in the penultimate column. In other situations, the row might define other characteristics of the column, such as the type of

---

[2] The following diagram is a simplified version of Figure 3 of the '604 patent.

data that the column can hold, e.g., text, integer numbers, or decimal numbers. Because the patent describes a model where the table's columns are defined by rows in that same table, it is "self-referential." *See* '604 patent, col. 2, ll. 59–65.

The patents teach that multiple benefits flow from this design. First, the patents disclose an indexing technique that allows for faster searching of data than would be possible with the relational model. *See, e.g.*, '604 patent, col. 1 ll. 55–59; *id.* at col. 2 l. 66–col. 3 l. 6. Second, the patents teach that the self-referential model allows for more effective storage of data other than structured text, such as images and unstructured text. *See, e.g.*, '604 patent, col. 2 ll. 16–22; col. 2 ll. 46–52.

Finally, the patents teach that the self-referential model allows more flexibility in configuring the database. *See, e.g.*, '604 patent, col. 2 ll. 27–29. In particular, whereas deployment of a relational database often involves extensive modeling and configuration of the various tables and relationships in advance of launching the database, Enfish argues that the self-referential database can be launched without such tasks and instead configured on-the-fly. *See* Oral Argument at 1:00–2:15 http://oralarguments.cafc.ucsourts.gov/default.aspx?fl=2015-1244.mp3; *see also* '604 patent, col. 7 ll. 10–22. For instance, the database could be launched with no or only minimal column definitions. Then, as a new attribute of information is encountered, such as an email address, an "Email" column could be added simply by inserting a new row of TYPE = "field" and LABEL = "email." The addition of this new row can then instigate the database to create a new, corresponding column. The addition of a new row-defining-a-column to the previous example might result in the following:

SELF-REFERENTIAL TABLE							
ID	**Type**	**Title**	**Label**	**Address**	**Employed By (#4)**	**Author**	**Email (#5)**
#1	DOCUMENT	PROJECT PLAN		C:\WORD\PROJ.DOC		#2	
#2	PERSON		SCOTT WLASCHIN		#3		
#3	COMPANY		DEXIS	117 EAST COLORADO			
#4	FIELD		EMPLOYED BY				
#5	FIELD		EMAIL				

In 2012, Enfish filed suit against Microsoft in district court in California, alleging that Microsoft's ADO.NET product infringes the '604 and '775 patents. ADO.NET provides an interface by which software applications can store, retrieve, and otherwise manipulate data stored in a database. Enfish alleges that ADO.NET creates and manipulates self-referential tables as part of its operation.

Five claims are at issue in this appeal: claims 17, 31, and 32 of the '604 patent; and claims 31 and 32 of the '775 patent. The district court entered summary judgment on these claims as follows: all claims invalid under 35 U.S.C. § 101 as directed to an abstract idea; claims 31 and 32 of both patents invalid under 35 U.S.C. § 102(b) as anticipated by the prior public sale and use of Microsoft's Excel 5.0 product; and claim 17 not infringed by ADO.NET.

Enfish appeals each of these summary judgments. We have jurisdiction under 28 U.S.C. § 1295(a)(1).

II

This court reviews a grant of summary judgment under the standard of review of the regional circuit. *See Taurus IP, LLC v. DaimlerChrysler Corp.*, 726 F.3d 1306, 1322 (Fed. Cir. 2013). The Ninth Circuit reviews a grant of summary judgment de novo. *See Oswalt v. Resolute Indus., Inc.*, 642 F.3d 856, 859 (9th Cir. 2011). Summary judgment is only appropriate if "there is no genuine dispute as to any material fact and the movant is entitled to judgment as a matter of law." FED. R. CIV. P. 56(a). In reviewing summary judgment, "[t]he evidence of the non-movant is to be believed and all justifiable inferences are to be drawn in [the non-movant's] favor." *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 255 (1986).

On appeal, Enfish challenges the district court's grant of summary judgment on § 101 invalidity, § 102 invalidi-

ty, and non-infringement. We address each argument in turn.

III

We turn first to the district court's determination that the claims at issue do not claim patent-eligible subject matter, which we review de novo. *See OIP Techs., Inc. v. Amazon.com, Inc.*, 788 F.3d 1359, 1362 (Fed. Cir. 2015).

Section 101 provides that a patent may be obtained for "any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof." 35 U.S.C. § 101. This court, as well as the Supreme Court, has long grappled with the exception that "[l]aws of nature, natural phenomena, and abstract ideas are not patentable." *Ass'n for Molecular Pathology v. Myriad Genetics, Inc.*, --- U.S. ----, 133 S. Ct. 2107, 2116 (2013) (quoting *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, --- U.S. ----, 132 S. Ct. 1289, 1293 (2012)). Supreme Court precedent instructs us to "first determine whether the claims at issue are directed to a patent-ineligible concept." *Alice Corp. Pty Ltd. v. CLS Bank Int'l*, --- U.S. ----, 134 S. Ct. 2347, 2355 (2014). If this threshold determination is met, we move to the second step of the inquiry and "consider the elements of each claim both individually and 'as an ordered combination' to determine whether the additional elements 'transform the nature of the claim' into a patent-eligible application." *Id.* (quoting *Mayo*, 132 S. Ct. at 1298, 1297).

The Supreme Court has not established a definitive rule to determine what constitutes an "abstract idea" sufficient to satisfy the first step of the *Mayo*/*Alice* inquiry. *See id.* at 2357. Rather, both this court and the Supreme Court have found it sufficient to compare claims at issue to those claims already found to be directed to an abstract idea in previous cases. "[The Court] need not labor to delimit the precise contours of the 'abstract ideas' category in this case. It is enough to recognize that there

is no meaningful distinction between the concept of risk hedging in *Bilski* and the concept of intermediated settlement at issue here." *Alice*, 134 S. Ct. at 2357; *see also OIP Techs.,* 788 F.3d at 1362. For instance, fundamental economic and conventional business practices are often found to be abstract ideas, even if performed on a computer. *See, e.g.*, *OIP Techs.*, 788 F.3d at 1362–63.

In setting up the two-stage *Mayo*/*Alice* inquiry, the Supreme Court has declared: "We must first determine whether the claims at issue are directed to a patent-ineligible concept." *Alice*, 134 S. Ct. at 2355. That formulation plainly contemplates that the first step of the inquiry is a meaningful one, i.e., that a substantial class of claims are *not* directed to a patent-ineligible concept. The "directed to" inquiry, therefore, cannot simply ask whether the claims *involve* a patent-ineligible concept, because essentially every routinely patent-eligible claim involving physical products and actions *involves* a law of nature and/or natural phenomenon—after all, they take place in the physical world. *See Mayo*, 132 S. Ct. at 1293 ("For all inventions at some level embody, use, reflect, rest upon, or apply laws of nature, natural phenomena, or abstract ideas.") Rather, the "directed to" inquiry applies a stage-one filter to claims, considered in light of the specification, based on whether "their character as a whole is directed to excluded subject matter." *Internet Patents Corp. v. Active Network, Inc.*, 790 F.3d 1343, 1346 (Fed. Cir. 2015); *see Genetic Techs. Ltd. v. Merial L.L.C.*, 2016 WL 1393573, at *5 (Fed. Cir. 2016) (inquiring into "the focus of the claimed advance over the prior art").

The Supreme Court has suggested that claims "purport[ing] to improve the functioning of the computer itself," or "improv[ing] an existing technological process" might not succumb to the abstract idea exception. *See Alice*, 134 S. Ct. at 2358–59. While it is true that the Court discussed improvements to computer-related technology in the second step of its analysis in *Alice*, *see id.* at

2355–60, that was because the Court did not need to discuss the first step of its analysis at any considerable length, *see id.* at 2356 ("Petitioner acknowledges that its claims describe intermediate settlement . . . ."), *id.* at 2357.

We do not read *Alice* to broadly hold that all improvements in computer-related technology are inherently abstract and, therefore, must be considered at step two. Indeed, some improvements in computer-related technology when appropriately claimed are undoubtedly not abstract, such as a chip architecture, an LED display, and the like. Nor do we think that claims directed to software, as opposed to hardware, are inherently abstract and therefore only properly analyzed at the second step of the *Alice* analysis. Software can make non-abstract improvements to computer technology just as hardware improvements can, and sometimes the improvements can be accomplished through either route. We thus see no reason to conclude that all claims directed to improvements in computer-related technology, including those directed to software, are abstract and necessarily analyzed at the second step of *Alice*, nor do we believe that *Alice* so directs. Therefore, we find it relevant to ask whether the claims are directed to an improvement to computer functionality versus being directed to an abstract idea, even at the first step of the *Alice* analysis.

For that reason, the first step in the *Alice* inquiry in this case asks whether the focus of the claims is on the specific asserted improvement in computer capabilities (i.e., the self-referential table for a computer database) or, instead, on a process that qualifies as an "abstract idea" for which computers are invoked merely as a tool. As noted *infra*, in *Bilski* and *Alice* and virtually all of the computer-related § 101 cases we have issued in light of those Supreme Court decisions, it was clear that the claims were of the latter type—requiring that the analysis proceed to the second step of the *Alice* inquiry, which asks

if nevertheless there is some inventive concept in the application of the abstract idea. *See Alice*, 134 S. Ct. at 2355, 2357–59. In this case, however, the plain focus of the claims is on an improvement to computer functionality itself, not on economic or other tasks for which a computer is used in its ordinary capacity.

Accordingly, we find that the claims at issue in this appeal are not directed to an abstract idea within the meaning of *Alice*. Rather, they are directed to a specific improvement to the way computers operate, embodied in the self-referential table. *See supra* at 6. Specifically, claim 17 of the '604 patent recites:

> A data storage and retrieval system for a computer memory, comprising:
>
> means for configuring said memory according to a logical table, said logical table including:
>
> a plurality of logical rows, each said logical row including an object identification number (OID) to identify each said logical row, each said logical row corresponding to a record of information;
>
> a plurality of logical columns intersecting said plurality of logical rows to define a plurality of logical cells, each said logical column including an OID to identify each said logical column; and
>
> means for indexing data stored in said table.

Pursuant to 35 U.S.C. § 112 ¶ 6 (2006), the district court construed the "means for configuring" language as requiring a four-step algorithm:[3]

> 1. Create, in a computer memory, a logical table that need not be stored contiguously in the computer memory, the logical table being comprised of rows and columns, the rows corresponding to records, the columns corresponding to fields or attributes, the logical table being capable of storing different kinds of records.
>
> 2. Assign each row and column an object identification number (OID) that, when stored as data, can act as a pointer to the associated row or column and that can be of variable length between databases.
>
> 3. For each column, store information about that column in one or more rows, rendering the table self-referential, the appending, to the logical table, of new columns that are available for immediate

---

[3] "Construction of a means-plus-function limitation includes two steps. 'First, the court must determine the claimed function. Second, the court must identify the corresponding structure in the written description of the patent that performs the function.'" *Noah Sys., Inc. v. Intuit Inc.*, 675 F.3d 1302, 1311 (Fed. Cir. 2012) (quoting *Applied Med. Res. Corp. v. U.S. Surgical Corp.*, 448 F.3d 1324, 1332 (Fed. Cir. 2006)). And "the corresponding structure for a function performed by a software algorithm is the algorithm itself." *EON Corp. IP Holdings LLC v. AT & T Mobility LLC*, 785 F.3d 616, 621 (Fed. Cir. 2015). The parties do not dispute this construction on appeal.

> use being possible through the creation of new column definition records.
>
> 4. In one or more cells defined by the intersection of the rows and columns, store and access data, which can include structured data, unstructured data, or a pointer to another row.

J.A. 325.

The district court concluded that the claims were directed to the abstract idea of "storing, organizing, and retrieving memory in a logical table" or, more simply, "the concept of organizing information using tabular formats." J.A. 321 (emphasis omitted). Likewise, Microsoft urges the court to view the claims as being directed to "the concepts of organizing data into a logical table with identified columns and rows where one or more rows are used to store an index or information defining columns." Appellee's Br. 17. However, describing the claims at such a high level of abstraction and untethered from the language of the claims all but ensures that the exceptions to § 101 swallow the rule. *See Alice*, 134 S. Ct. at 2354 (noting that "we tread carefully in construing this exclusionary principle [of laws of nature, natural phenomena, and abstract ideas] lest it swallow all of patent law"); *cf. Diamond v. Diehr*, 450 U.S. 175, 189 n.12 (1981) (cautioning that overgeneralizing claims, "if carried to its extreme, make[s] all inventions unpatentable because all inventions can be reduced to underlying principles of nature which, once known, make their implementation obvious").

Here, the claims are not simply directed to *any* form of storing tabular data, but instead are specifically directed to a *self-referential* table for a computer database. For claim 17, this is reflected in step three of the "means for configuring" algorithm described above. For both pairs of claims 31 and 32, this is reflected in other claim language, discussed *infra* at 20. The necessity of describ-

ing the claims in such a way is underscored by the specification's emphasis that "the present invention comprises a flexible, self-referential table that stores data." '604 patent at Abstract; *see also id.* at col. 2 ll. 44–46 ("The present invention improves upon prior art information search and retrieval systems by employing a flexible, self-referential table to store data.").

The specification also teaches that the self-referential table functions differently than conventional database structures. According to the specification, traditional databases, such as "those that follow the relational model and those that follow the object oriented model," '604 patent, col. 1 ll. 37–40, are inferior to the claimed invention. While "[t]he structural requirements of current databases require a programmer to predefine a structure and subsequent [data] entry must conform to that structure," *id.* at col. 2 ll. 10–13, the "database of the present invention does not require a programmer to preconfigure a structure to which a user must adapt data entry." *Id.* at col 2 ll. 27–29. Moreover, our conclusion that the claims are directed to an improvement of an existing technology is bolstered by the specification's teachings that the claimed invention achieves other benefits over conventional databases, such as increased flexibility, faster search times, and smaller memory requirements. *See id.* at col 2 ll. 23–27; *see also Openwave Sys., Inc. v. Apple Inc.*, 808 F.3d 509, 513–14 (Fed. Cir. 2015) (finding that a specification's disparagement of the prior art is relevant to determine the scope of the invention).

In finding that the claims were directed simply to "the concept of organizing information using tabular formats," J.A. 321 (emphasis omitted), the district court oversimplified the self-referential component of the claims and downplayed the invention's benefits. The court determined that the patents' self-referential concept could be satisfied by creating a table with a simple header row. But that is simply not the case. For example, step three

of the algorithm described above explains that the table stores information related to each column in rows of that very same table, such that new columns can be added by creating new rows in the table. *See* J.A. 325 (describing four-step algorithm); *see also* '604 patent, col. 2 ll. 53–65 (describing "the present invention", including a description where "columns are entered as rows in the table and the record corresponding to a column contains various information about the column," thereby "render[ing] the table self-referential"). It is beyond debate that this is more than simply a header row.

Moreover, we are not persuaded that the invention's ability to run on a general-purpose computer dooms the claims. Unlike the claims at issue in *Alice* or, more recently in *Versata Development Group v. SAP America, Inc.,* 793 F.3d 1306 (Fed. Cir. 2015), which Microsoft alleges to be especially similar to the present case, Appellee's Br. 18, *see also* Oral Argument at 15:40–18:15, the claims here are directed to an improvement in the functioning of a computer. In contrast, the claims at issue in *Alice* and *Versata* can readily be understood as simply adding conventional computer components to well-known business practices. *See Alice*, 134 S. Ct. at 2358–60; *Versata Dev. Grp.*, 793 F.3d at 1333–34 (computer performed "purely conventional" steps to carry out claims directed to the "abstract idea of determining a price using organization and product group hierarchies"); *see also Mortgage Grader, Inc. v. First Choice Loan Servs. Inc.*, 811 F.3d 1314, 1324–25 (Fed. Cir. 2016) (claims attaching generic computer components to perform "anonymous loan shopping" not patent eligible); *Intellectual Ventures I LLC v. Capital One Bank (USA)*, 792 F.3d 1363, 1367–69 (Fed. Cir. 2015) (claims adding generic computer components to financial budgeting); *OIP Techs.,* 788 F.3d at 1362–64 (claims implementing offer-based price optimization using conventional computer activities); *Ultramercial, Inc. v. Hulu, LLC*, 772 F.3d 709, 714–17 (Fed. Cir.

2014) (claims applying an exchange of advertising for copyrighted content to the Internet); *buySAFE, Inc. v. Google, Inc.*, 765 F.3d 1350, 1354–55 (Fed. Cir. 2014) (claims adding generic computer functionality to the formation of guaranteed contractual relationships). And unlike the claims here that are directed to a specific improvement to computer functionality, the patent-ineligible claims at issue in other cases recited use of an abstract mathematical formula on any general purpose computer, *see Gottschalk v. Benson*, 409 U.S. 63, 93 (1972), *see also Alice*, 134 S. Ct. at 2357–58, or recited a purely conventional computer implementation of a mathematical formula, *see Parker v. Flook*, 437 U.S. 584, 594 (1978); *see also Alice*, 134 S. Ct. at 2358, or recited generalized steps to be performed on a computer using conventional computer activity, *see Internet Patents*, 790 F.3d 1348–49 (claims directed to abstract idea of maintaining computer state without recitation of specific activity used to generate that result), *Digitech Image Techs., LLC v. Electrs. For Imaging, Inc.*, 758 F.3d 1344, 1351 (Fed. Cir. 2014) (claims directed to abstract idea of "organizing information through mathematical correlations" with recitation of only generic gathering and processing activities).

Similarly, that the improvement is not defined by reference to "physical" components does not doom the claims. To hold otherwise risks resurrecting a bright-line machine-or-transformation test, *cf. Bilski v. Kappos*, 561 U.S. 593, 604 (2010) ("The machine-or-transformation test is not the sole test for deciding whether an invention is a patent-eligible 'process.'"), or creating a categorical ban on software patents, *cf. id.* at 603 ("This Court has not indicated that the existence of these well-established exceptions gives the Judiciary *carte blanche* to impose other limitations that are inconsistent with the text and the statute's purpose and design."). Much of the advancement made in computer technology consists of improvements to

software that, by their very nature, may not be defined by particular physical features but rather by logical structures and processes. We do not see in *Bilski* or *Alice*, or our cases, an exclusion to patenting this large field of technological progress.

In sum, the self-referential table recited in the claims on appeal is a specific type of data structure designed to improve the way a computer stores and retrieves data in memory. The specification's disparagement of conventional data structures, combined with language describing the "present invention" as including the features that make up a self-referential table, confirm that our characterization of the "invention" for purposes of the § 101 analysis has not been deceived by the "draftsman's art." *Cf. Alice,* 134 S. Ct. at 2360. In other words, we are not faced with a situation where general-purpose computer components are added post-hoc to a fundamental economic practice or mathematical equation. Rather, the claims are directed to a specific implementation of a solution to a problem in the software arts. Accordingly, we find the claims at issue are not directed to an abstract idea.

Because the claims are not directed to an abstract idea under step one of the *Alice* analysis, we do not need to proceed to step two of that analysis. *See id.* at 2355. We recognize that, in other cases involving computer-related claims, there may be close calls about how to characterize what the claims are directed to. In such cases, an analysis of whether there are arguably concrete improvements in the recited computer technology could take place under step two. Here, though, we think it is clear for the reasons stated that the claims are not directed to an abstract idea, and so we stop at step one. We conclude that the claims are patent-eligible.

IV

Alternatively, Microsoft encourages us to affirm the invalidity of claim 17 on the ground of indefiniteness.

According to Microsoft, the previously-recited four-step algorithm is not a sufficient structure for the claimed function of "configuring said memory according to a logical table."

For a claim element recited in means-plus-function format, "the specification must contain sufficient descriptive text by which a person of skill in the field of the invention would 'know and understand what structure corresponds to the means limitation.'" *Typhoon Touch Techs., Inc. v. Dell, Inc.*, 659 F.3d 1376, 1383–84 (Fed. Cir. 2011) (quoting *Finisar Corp. v. DirecTV Grp., Inc.*, 523 F.3d 1323, 1340 (Fed. Cir. 2008)). "[W]hile it is true that the patentee need not disclose details of structures well known in the art, the specification must nonetheless disclose some structure." *Biomedino LLC v. Waters Techs. Corp.*, 490 F.3d 946, 952 (Fed. Cir. 2007) (quoting *Default Proof Credit Card Sys. v. Home Depot U.S.A., Inc.*, 412 F.3d 1291, 1302 (Fed. Cir. 2005)).

The district court found that the four-step algorithm sufficiently identified a structure for a person of skill in the art to implement the function of "configuring said memory according to a logical table." We agree. Step one of the four-step algorithm relies on well-known techniques in the database arts for setting up a table in computer memory. Microsoft does not allege that an ordinary artisan would not understand the algorithm. Steps two through four then provide particular details for modifying some such well-known configuration in accordance with the disclosed invention. The fact that this algorithm relies, *in part*, on techniques known to a person of skill in the art does not render the composite algorithm insufficient under § 112 ¶ 6. Indeed, this is entirely consistent with the fact that the sufficiency of the structure is viewed through the lens of a person of skill in the art and without need to "disclose structures well known in the art," *Biomedino*, 490 F.3d at 952.

Therefore, we do not find claim 17 invalid on this alternative ground.

V

Because we find the claims patent-eligible under § 101, we now turn to the issue of validity under § 102. The district court found claims 31 and 32 of both patents anticipated under § 102. Claim 31 of the '604 patent is exemplary:

> A method for storing and retrieving data in a computer memory, comprising the steps of:
>
> configuring said memory according to a logical table, said logical table including:
>
> a plurality of logical rows, each said logical row including an object identification number (OID) to identify each said logical row, each said logical row corresponding to a record of information;
>
> a plurality of logical columns intersecting said plurality of logical rows to define a plurality of logical cells, each said logical column including an OID to identify each said logical column; and
>
> wherein at least one of said logical rows has an OID equal to the OID to a corresponding one of said logical columns, and at least one of said logical rows includes logical column information defining each of said logical columns.

As seen, claim 31 recites a method involving configuring a memory "according to a logical table," with that logical table specified in some detail. Notably, the logical table must have a row and a column that have the same ID value. This language of the "wherein" clause embodies the self-referential property explained *supra* at 6.

The district court found the claims anticipated under § 102(b) by the Microsoft Excel 5.0 software program. Excel 5.0 is a version of the well-known spreadsheet program that was in public use at latest by early-1994. The district court focused on the "pivot table" feature of Excel 5.0 as anticipating the claims. A pivot table is a type of data-summarization table that a user can prompt the Excel program to generate, based on a preexisting table of raw data. For instance, a user may begin with a table of raw sales data as follows (only a subset of the entire table is shown below):

Product	Year	Month	Sales	Units	Salespers	Region
Dairy	1992	Dec	7686	5563	Davolio	North
Produce	1993	Sep	2956	1242	Buchanan	West
Produce	1992	Oct	8165	983	Buchanan	South
Dairy	1993	Jan	4448	3833	Buchanan	North
Dairy	1993	Sep	75	3216	Buchanan	East
Produce	1993	Feb	4923	8160	Davolio	South
Dairy	1993	Dec	2733	2790	Davolio	West

J.A. 7722. The example table has rows for sales by a particular salesperson of a particular product in a particular region, among other attributes. The user can prompt Excel to create a pivot table, such as the following:

Sum of Sales	Product		
Salesperson	Dairy	Produce	Grand Total
Buchanan	67566	59259	126825
Davolio	87916	71829	159745
Grand Total	155482	131088	286570

J.A. 7722. This pivot table has row labels corresponding to salespersons and column labels corresponding to types of product. The cells of the pivot table sum the "Sales" column of the raw data table based on its intersection of a particular salesperson and a particular type of product.

For instance, salesperson Buchanan has sold $67,566 worth of dairy products.

The district court found this pivot table feature of Excel 5.0 to anticipate claim 31, including the self-referential property embodied in the language of "wherein at least one of said logical rows has an OID equal to the OID to a corresponding one of said logical columns." The district court noted that a cell in the row of the raw data table, (e.g., "Dairy") was also the label of a column in the pivot table, (again, "Dairy"). J.A. 283–84. Microsoft's expert exemplified this position by showing that the addition of the "Housewares" row to the raw data table, like so:

Product	Year	Month	Sales	Units	Salespers	Region
Dairy	1992	Dec	7686	5563	Davolio	North
Produce	1993	Sep	2956	1242	Buchanan	West
Produce	1992	Oct	8165	983	Buchanan	South
Produce	1992	Jun	1361	1824	Davolio	South
Dairy	1992	Apr	9136	2021	Buchanan	East
Housewares	1995	Apr	1000	10000	Gray	West

J.A. 7723, would result in the addition of a "Housewares" column to the pivot table, like so:

Sum of Sales	Product			
Salesperson	Dairy	Produce	Housewares	Grand Total
Buchanan	67566	59259	0	126825
Davolio	87916	71829	0	159745
Gray	0	0	1000	1000
Grand Total	155482	131088	1000	287570

J.A. 7724. On its face, this would seem to have the effect of adding a column based on a newly added row, which is in some ways a characteristic behavior of the self-referential table disclosed in the patents.

But finding this feature to anticipate the claims requires an inappropriately broad reading of the claims. Claim 31 is directed to configuring memory "according to

a logical table." While we have held that the use of a singular indefinite article with a claim feature may support an interpretation of "one or more" of those claim features, *see, e.g.*, *Free Motion Fitness, Inc. v. Cybex Int'l, Inc.*, 423 F.3d 1343, 1350 (Fed. Cir. 2005), the context of claim 31 shows that this is not such a case. The remainder of claim 31 describes rows and columns without providing any suggestion that a second table has been introduced. The specification makes clear that the invention is directed to the arrangement of a single, logical table, particularly, a row defining a column *in that same table*. *See, e.g.*, '604 patent, col. 2, ll. 31–41; *id.* at col. 2, ll. 44–52; *id.* at col. 7 ll. 10–22; *id.* at Figure 3; *see also* Oral Argument at 1:45–2:15; *id.* at 27:00–29:30. Therefore, the "at least one of said logical rows" and the "corresponding one of said logical columns" must both be in the same logical table.[4]

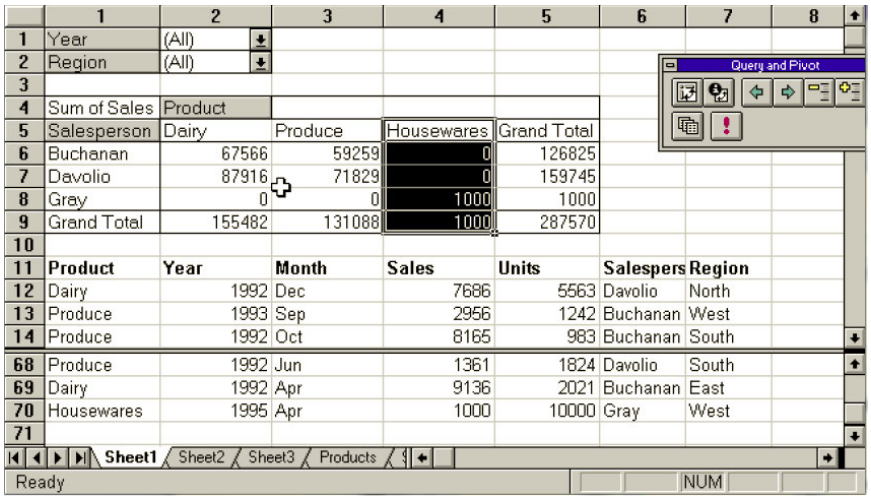
But the district court read the features of claim 31 on a row from the raw data table and a column from the pivot table. This fails to show the feature of claim 31 having identical IDs for a row and a column in the same table. Fundamentally, having a row in one table reference a column in another table is not a "***self***-referential" table at all, but something more like "referential tables" or "tables that refer to one another."

The fact that the raw data table and the pivot table are present on the same spreadsheet is of no consequence. The district court appears to have grounded its reasoning

---

[4] To be clear, we do not hold that the claims are directed exclusively to a database with a single, self-referential table. Rather, the claims recite a single, self-referential table, regardless of any other tables that may be present in the same database.

on the fact that the two tables show up together on a single spreadsheet:



	1	2	3	4	5	6	7	8
1	Year	(All)						
2	Region	(All)						
3								
4	Sum of Sales	Product						
5	Salesperson	Dairy	Produce	Housewares	Grand Total			
6	Buchanan	67566	59259	0	126825			
7	Davolio	87916	71829	0	159745			
8	Gray	0	0	1000	1000			
9	Grand Total	155482	131088	1000	287570			
10								
11	**Product**	**Year**	**Month**	**Sales**	**Units**	**Salespers**	**Region**	
12	Dairy	1992	Dec	7686	5563	Davolio	North	
13	Produce	1993	Sep	2956	1242	Buchanan	West	
14	Produce	1992	Oct	8165	983	Buchanan	South	
68	Produce	1992	Jun	1361	1824	Davolio	South	
69	Dairy	1992	Apr	9136	2021	Buchanan	East	
70	Housewares	1995	Apr	1000	10000	Gray	West	
71								

J.A. 7724. But the spreadsheet is no more than the medium on which the two separate tables are presented. Two separate tables drawn on one sheet of paper are still two separate tables.

Therefore, Excel 5.0 fails to include the claimed single table having a row defining a column in that same table. Identification of one element, the row, in one table and another element, the column, in another table is insufficient for anticipation. Anticipation requires "that the reference describe not only the elements of the claimed invention, but also that it describe those elements 'arranged as in the claim[.]'" *Net MoneyIN, Inc. v. VeriSign, Inc.*, 545 F.3d 1359, 1371 (Fed. Cir. 2008) (quoting *Finisar Corp. v. DirecTV Grp., Inc.*, 523 F.3d 1323, 1334 (Fed. Cir. 2008)).

For this reason, the district court erred in granting summary judgment of anticipation. Given our claim construction and the disclosure in Excel 5.0, the pivot table feature of Excel 5.0 does not anticipate claim 31 of

the '604 patent. Claim 32 of the '604 patent and claims 31 and 32 of the '775 patent require the same "self-referencing" feature by way of the matching-ID language. Therefore, we find that the pivot table feature of Excel 5.0 also does not anticipate those claims.

***

Enfish encourages us to consider whether Microsoft is estopped from asserting an Excel 5.0 invalidity defense due to an *inter partes* review instituted at the U.S. Patent and Trademark Office at Microsoft's request. Because we find that Excel 5.0 does not anticipate the claims, we see no reason to address this issue. If on remand the district court permits a new invalidity contention based on a different feature of Excel 5.0 or some other related prior art, then we leave the estoppel issue to the district court to consider in the first instance.

VI

We now turn to the issue of infringement. The district court found that Microsoft's accused product, ADO.NET, does not infringe claim 17 of the '604 patent. The district court reached this conclusion by finding that ADO.NET does not perform the "means for indexing" recited in that claim. Claim 17, in abbreviated form, recites as follows:

> A data storage and retrieval system for a computer memory, comprising:
>
> means for configuring said memory according to a logical table . . . and
>
> means for indexing data stored in said table.

Enfish raises two arguments against the district court's summary judgment of non-infringement: first against the claim construction for "means for indexing," and second for the application of that claim construction to the ADO.NET product.

A

The district court interpreted the "means for indexing" under 35 U.S.C. § 112 ¶ 6 (2006). Such a claim element "shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof." § 112 ¶ 6. And, as noted above, "the corresponding structure for a function performed by a software algorithm is the algorithm itself." *EON*, 785 F.3d at 621. The district court identified the function of the "means for indexing" as "indexing data stored in the logical table." J.A. 270. The district court accepted Enfish's proposal of the following algorithm as the corresponding structure:

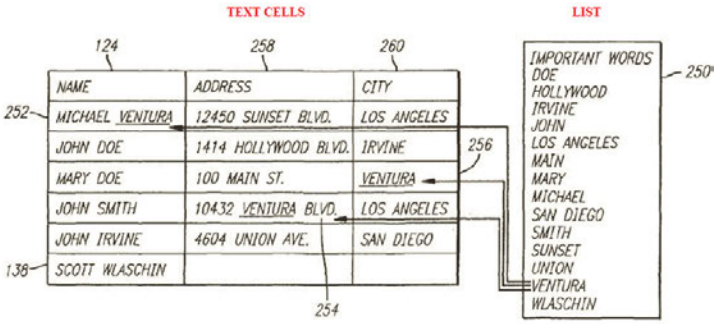
1. Extract key phrases or words from the applicable cells in the logical table.
2. Store the extracted key phrases or words in an index, which is itself stored in the logical table.
3. Include, in text cells of the logical table, pointers to the corresponding entries in the index, and include, in the index, pointers to the text cells.

J.A. 270, 338, 2543–49.

On appeal, Enfish now contests the district court's reliance on this three-step algorithm as the corresponding structure for the "means for indexing." Enfish argues that it never meant for all three steps to be required, but instead that some of the steps or parts of the steps may be optional. Specifically, Enfish argues that the corresponding structure does not necessarily require both "pointers to the . . . index" and "pointers to the text cells" in step 3, i.e., "bi-directional" pointers. Appellant's Br. 61–63. Enfish asserts that because the specification shows both an embodiment with uni-directional pointers and an embodiment with bi-directional pointers, the district court erred in identifying the three-step algorithm with bi-directional pointers as the only corresponding structure.

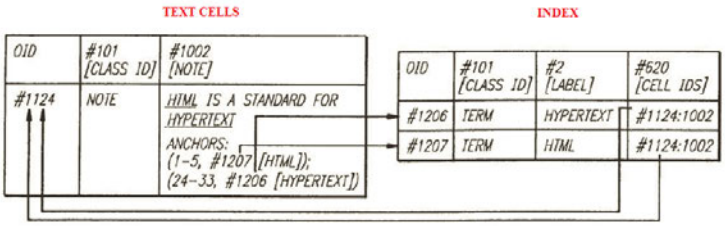
The district court did not err in its construction. Enfish's primary support for its position is the fact that Figure 11 of the patents shows uni-directional pointers (i.e., from the index to the text cells), whereas Figure 14 shows bi-directional pointers (i.e., in both directions between the index and the text cells).

FIG. 11



'604 patent, Figure 11 ("TEXT CELLS" and "LIST" markup our own).

FIG. 14



'604 patent, Figure 14 ("TEXT CELLS" and "INDEX" markup our own). Although Figures 11 and 14 do in fact illustrate a sort of dichotomy akin to that proffered by Enfish, the discussion of those figures in the specification reveals that any such dichotomy is false. The list 250 in Figure 11 is not an index at all; it is an intermediate compilation of keyword values used in the process of forming the index. *See generally* '604 patent, col. 12–14. The specification notes that, even at the stage of forming list 250, the text cells contain references to the keywords

contained therein, in the form of "anchors." *Id.* at col. 12 ll. 16–34. When the index is ultimately formed, as illustrated in Figure 14, the index entries contain references to the text cells, and the text cells, through the use of the anchors, contain references to the index. *See id.* at col. 14 ll. 10–17. That is, the specification presents Figure 14 as index entries in the self-referential table that result from a process that began with the keyword extraction step illustrated in Figure 11. The figures are not alternative embodiments.

Therefore, we find no error in the district court's use of Enfish's own-identified three-step algorithm as the sole equivalent structure for the "means for indexing."

B

Using the three-step algorithm, the district court determined that ADO.NET does not perform either step two or step three. For step two, the district court determined that there was no genuine issue of fact that ADO.NET does not store the text value of a keyword in the index, but rather just a reference to that value. For step three, the district court determined that there was no genuine issue of fact that ADO.NET does not store a pointer from the text value to the index, but stores a pointer to some other object. Enfish argues that the district court erred in its findings as to both step two and step three.

For an accused product to practice a claim element interpreted under § 112 ¶ 6, the accused product must perform the identical function using an identical or equivalent structure. *See Odetics, Inc. v. Storage Tech. Corp.*, 185 F.3d 1259, 1267 (Fed. Cir. 1999). Step two requires storing the "extracted key phrases or words" in an index. However, as the district court explained, it is uncontested that ADO.NET does not store the actual text values in the index, but rather stores references to the cells containing those text values. J.A. 340–41; Appellant's Br. 66. Step three requires including, in the cells with the text values,

pointers to the "corresponding entries in the index." Again, it is uncontested that ADO.NET does not store pointers as such. Instead, it stores pointers to other objects that, by use of a further chain of pointers, may be resolved to the index. J.A. 341–43; Appellant's Br. 69. Therefore, ADO.NET does not use the identical "corresponding structure," i.e., the three-step algorithm, disclosed in the specification.

As such, ADO.NET can only infringe claim 17 if its algorithm for performing indexing is an equivalent to the three-step algorithm identified above. An accused structure is "equivalent" to a disclosed structure if the differences between the two are insubstantial. *See Odetics*, 185 F.3d at 1267. The district court concluded that the failure to store actual text values in the index (i.e., the difference at step two) combined with the failure to use pointers from the text values to the index (i.e., the difference at step three) render the three-step algorithm and the ADO.NET indexing algorithm substantially different. J.A. 343–44. The district court found that at least one disclosed organization scheme for the index, i.e., alphabetically for fast name searching, may not be possible if the actual text values are not stored in the index. The district court found that the bi-directional pointers allow the performance of associative queries, which the patents describe as being a key benefit of the invention. On appeal, Enfish attempts to argue that the differences in step two and step three do not exist, but fails to explain how those differences are insubstantial if we agree with the district court that they do exist, which we do. Therefore, finding no argument to the contrary, we conclude that ADO.NET's indexing algorithm is not an equivalent of the three-step algorithm of claim 17.

Because ADO.NET does not use the identical or equivalent structure as disclosed in the patents for the "means for indexing," we find that ADO.NET does not infringe claim 17.

VII

Based on the foregoing, we reverse the district court's grant of summary judgment based on § 101 and conclude that all five claims on appeal are patent-eligible. We vacate the district court's grant of summary judgment based on § 102 and conclude that both pairs of claims 31 and 32 are not anticipated by Excel 5.0 pivot tables. Lastly, we affirm the district court's grant of summary judgment of non-infringement and conclude that ADO.NET does not infringe claim 17. We remand the case to the district court for further proceedings.

REVERSED-IN-PART, VACATED-IN-PART, AFFIRMED-IN-PART, AND REMANDED

No costs.