IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE PATENT TRIAL AND APPEAL BOARD

U.S. Patent No. 6,529,908B1

Issued: March 4, 2003

Named Inventors: Piett et al.

Original Assignee: Netspan Corporation; Current Assignee: Data Distribution Technologies LLC

Filed: May 28, 1998

Title: **WEB-UPDATED DATABASE WITH RECORD DISTRIBUTION BY EMAIL**

Petition filed:
May 19, 2016

Filed Electronically per
37 CFR § 42.6(b)(1)

Trial Number: [To Be Assigned]

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 6,529,908B1
UNDER 35 U.S.C. §§ 311-319 AND 37 C.F.R. § 42.100 et seq.

Pursuant to 35 U.S.C. §§ 311-319 and 37 C.F.R. § 42.100, the undersigned, on behalf of and acting in a representative capacity for petitioner National Association of Realtors ("Petitioner"), hereby petitions for *Inter Partes* review of Claims 1-63, 77, 88, 99, and 100 of U.S. Patent No. 6,529,908B1 ("the '908 Patent").

TABLE OF CONTENTS

Page

LIST OF EXHIBITS

Exhibit No.	Exhibit Description
1001	U.S. Patent No. 6,529,908B1
1002	The prosecution history of U.S. Patent No. 6,529,908B1
1003	Claim Equivalence Chart
1004	Declaration Under 37 CFR § 42.53 of Don Turnbull, Ph.D.
1005	*Curriculum Vitae* of Don Turnbull, Ph.D.
1006	Peterson, Erik S., *Building Applications with WebHub, Web Techniques Magazine* (July 1997)
1007	Jensen, Cary, *Delphi in Depth* (1996), Chapters 17, 19, and 20
1008	Papaj, Robert, *Oracle Databases on the Web Learn to Create Web Pages That Interface with Database Engines* (1997), Chapters 5, 7, and 17
1009	Surfas, Mark, *Running a Perfect Web Site with Windows* (1996), Chapters 5, 9, 13 and 15

Exhibit No.	**Exhibit Description**
1010	Chapman, Davis, *Building Internet Applications with Delphi 2* (1996), Chapters 8, 9, 17, and 12 and Appendix E
1011	Kersnar, Scott, *NetSuccess How Real Estate Agents Use the Internet* (1996), Chapters 1, 3 and 4
1012	Microsoft Press Computer Dictionary, 3rd Edition (1997)
1013	Calvert, Charles, *Delphi 2 Unleashed* (1996), inside cover page
1014	Declaration Under 37 CFR § 42.53 of Dr. Philip Greenspun
1015	Resumé of Dr. Philip Greenspun
1016	U.S. Patent No. 6,173,284
1017	Gladney, H.M., *"Data Replicas in Distributed Information Services,"* ACM Transactions on Database Systems 14:1 (March 1989)
1018	U.S. Patent 5,548,753

Exhibit No.	**Exhibit Description**
1019	U.S. Patent 5,974,396
1020	Texas Instruments, C++ Object-Oriented Library User's Manual (March 1990)
1021	Business Wire, *JobDirect.com Revolutionizes College Job Search With New Products and Industry Firsts* (November 2, 1998 and January 7, 1999)
1022	Information Sciences Institute, *Internet Protocol, DARPA Internet Program Protocol Specification* (September 1981)
1023	Codd, E.F., *A Relational Model of Data for Large Shared Data Banks,* Communications of the ACM, 13:6, June 1970
1024	*System R: An architectural overview*, M.W. Blasgen, et al., IBM Syst J, 20:1 (1981)
1025	Bowman, Judith S., et al., *The Practical SQL Handbook Using Structured Quary Language,* Third Edition (1996)

Exhibit No.	**Exhibit Description**
1026	The Curriculum Development Council Recommended for Use in Schools by The Education Department Hong Kong, *Syllabuses for Secondary Schools, Computer Applications (Advanced Supplementary Level)* (1998)
1027	Saltzer, Jerome H., *Protection and the Control of Information Sharing in Multics*, Communications of the ACM, 17:7 (July 1974)
1028	Information Sciences Institute, *Internet Protocol, Darpa Internet Program Protocol Specification* (September 1981)
1029	https://wiki.cc.gatech.edu/folklore/index.php/UNIVAC_I_Instruction_Set
1030	Baker, Henry G., *Efficient Implementation of Bit-Vector Operations in Common Lisp*, Henry G. Baker, (1990); ACM SIGPLAN Lisp Pointers, Volume III Issue 2-4, pp. 8-22
1031	Kernighan, Brian W. and Ritchie, Dennis M., *The C Programming Language*, Second Edition (1988)

Exhibit No.	**Exhibit Description**
1032	Posting in the newsgroup Microsoft.public.sqlserver.programming, October 7, 1997
1033	*R*: An Overview of the Architecture,* IBM Research, Almaden Research Center, Readings in Database Systems, Second Edition (1994)
1034	*Computation and Communication in R*: A Distributed Database Manager*, Lindsay, Bruce G., et al, ACM Transactions on Computer Systems, 2-1 (February 1984)
1035	*Updating Distributed Materialized Views,* Segev, Arie and Park, Jooseok IEEE Transactions on Knowledge and Data Engineering, 1:2 (June 1989)
1036	*Achieving Interoperability with Distributed Relational Databases,* Aiken, John A., et al., January 1991 IEEE Network Magazine
1037	*Mariposa: A New Architecture for Distributed Data*, Stonebraker, M., et al., IEEE 1994
1038	*Data Replication in Mariposa*, Sidell, J., et al., 1006 IEEE

Exhibit No.	Exhibit Description
1039	Pew Research Center, *The Internet Circa 1998* (June 21, 2007)
1040	*E-Mail Alerts Showing Growing Potential*, Bill Dedman, New York Times, Business Day, February 9, 1998
1041	*Rover: A Toolkit for Mobile Information Access*, Joseph, Anthony, D., et al., M.I.T. Laboratory for Computer Science, (1995)
1042	*World Wide Web Access to DB2*, IBM International Support Organization (May 1996)
1043	*World-Wide Web: The Information Universe*, T. Berners-Lee, et al., Electronic Networking, 2:1 pp. 52-58 (Spring 1992)
1044	Adams, David and Beckett, Dan, *Programming 4th Dimension, The Ultimate Guide* (January 1998)
1045	Language Tips, C++ Toolbox, *Evolution of the C++ Standard Library*, Dodgson, David, ACM SIGPLAN Notices, Vol. 31, Issue 12 (December 12, 1996)

Exhibit No.	**Exhibit Description**
1046	Business Wire, *Triggers now available for Informix database servers* (November 10, 1992)
1047	U.S. Patent 6,249,291B1
1048	*Data Distribution Technologies LLC v. Brer Affiliates, Inc. and Prudential Financial, Inc.*, United States District Court, District of New Jersey, Civil Action No. 1:12-cv-4878-JBS-KMW, Dkt. 41

I. INTRODUCTION

The challenged claims simply restate and combine old and well-known database fundamentals. The claims comprise obvious predictable combinations of foundational database concepts existing and in use decades prior. In fact, all the features and functionality claimed as innovative in the ’908 Patent were readily available in inexpensive, off-the-shelf, retail software products. In view of the prior art identified below, Petitioner respectfully requests a determination that all challenged claims are obvious and unpatentable.

II. MANDATORY NOTICES

Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner provides the following mandatory disclosures.

A. The Real Party-in-Interest

Petitioner certifies that National Association of Realtors is the real party-in-interest.

B. Related Matters

The ’908 Patent was originally assigned to Netspan Corporation. Netspan Corporation subsequently assigned it to Data Distribution Technologies LLC (“Patent Owner”). According to their own website (http://www.datadistributiontech.com/) and publicly available records, Patent Owner has filed six patent infringement lawsuits asserting the ’908 Patent:

1) *Data Distribution Technologies LLC v. Brer Affiliates Inc. and Prudential Financial, Inc*. (1:2012cv04978) filed in U.S. District Court for the District of New Jersey in 2012. According to Patent Owner, the case settled in 2013.

2) *Data Distribution Technologies LLC v. RE/MAX LLC* (1:2012cv04877) filed in U.S. District Court for the District of New Jersey in 2012. According to Patent Owner, the case settled and RE/MAX took a license under the ’908 Patent.

3) *Data Distribution Technologies LLC v. Weichert Real Estate Affiliates, Inc.* (2:12-cv-04149) filed in U.S. District Court for the District of New Jersey in 2012. According to Patent Owner, the case settled in 2013.

4) *Data Distribution Technologies LLC v. Pricegrabber.com Inc.* (7:11-cv-09650) filed in U.S. District Court for the Southern District of New York in 2011. According to Patent Owner, the case settled and Pricegrabber took a license under the ’908 Patent.

5) *Data Distribution Technologies LLC v. Kayak Software Corporation* (7:11-cv-09651) filed in U.S. District Court for the Southern District of New York in 2011. According to Patent Owner, the case settled in 2013.

6) *Data Distribution Technologies LLC v. Zoolert LLC* (7:11-cv-09653) filed in U.S. District Court for the Southern District of New York in 2011. According to Patent Owner, the case settled and Zoolert took a license under the ’908 Patent.

Petitioner is not aware of any other pending matters related to the '908 Patent.

C. Lead and Back-up Counsel

Pursuant to 37 C.F.R. § 42.8(b)(3), Petitioner provides the following designation of counsel: Lead counsel is Richard T. Black (Registration No. 40,514), and back-up counsel is Joel B. Ard (Registration No. 67,041), P.G. Scott Born (Registration No. 40,523), Benjamin J. Hodges (Registration No. 69,500), and Yeu-Yan Perng (Registration No. 74,815).

D. Service Information

37 C.F.R. § 42.8(b)(4), papers concerning this matter should be served on the following:

Address:	Richard T. Black Joel B. Ard P.G. Scott Born Benjamin J. Hodges Yeu-Yan Perng FOSTER PEPPER PLLC 1111 THIRD AVENUE, SUITE 3000 SEATTLE, WA 98101
Email:	Rich.Black@foster.com Joel.Ard@foster.com Scott.Born@foster.com Ben.Hodges@foster.com Yan.Perng@foster.com
Telephone:	(206) 447-4400
Facsimile:	(206) 749-2062

Petitioners consent to electronic service by email. Pursuant to 37 C.F.R. § 42.10(b), a Power of Attorney accompanies this Petition.

III. PAYMENT OF FEES

Requisite fees, including additional fees for challenging 47 claims over 20 (67 total), electronically accompany this Petition. The undersigned also authorizes the Patent Trial and Appeal Board ("PTAB") to charge Deposit Account No. 061629 for any fees required by 37 C.F.R. § 42.15(a) for this Petition for *Inter Partes* Review. The undersigned further authorizes payment for any additional fees and any overpayment of fees be credited to the above-referenced Deposit Account.

IV. REQUIREMENTS FOR INTER PARTES REVIEW

A. Grounds for Standing

Petitioner certifies per 37 C.F.R. § 42.104(a) the '908 Patent is available for, and the Petitioner is not barred or estopped from requesting, *Inter Partes* review challenging these claims of the '908 Patent. The '908 Patent has not been subject to a previous estoppel-based proceeding of the AIA, and no complaint has been served on Petitioner.

B. Identification of Challenge

Per 37 C.F.R. § 42.104(b) and (b)(1), Petitioner requests *Inter Partes* review of claims 1-63, 77, 88, 99, and 100 of the ’908 Patent, and that the PTAB declares them unpatentable.

C. Principal Grounds of Unpatentability Under 35 U.S.C. § 103

Gr.#	Reference	Claims
1	Peterson, Jensen	1{22,42,62}, 2{23,43}, 21, 63{77,88,99}, and 100
2	Peterson, Jensen, Papaj	3-7{24-28,44-48} and 14-20{35-41,55-61}
3	Peterson, Jensen, Papaj, Surfas, Chapman	8-13{29-34,49-54}
4	Brown, Linstead, Gladney	1{22,42,62},2{23,43}, 21,63{77,88,99} and 100
5	Brown, Linstead, Gladney, Anderson	3-17{24-38,44-59}
6	Brown, Linstead, Gladney, Anderson, C++Library	18-20{39-41,59-61}

1. The Specific Art and Statutory Grounds on Which the Challenge is Based

Pursuant to 37 C.F.R. § 42.204(b)(2), *Inter Partes* review of the '908 Patent is requested in view of the following 35 U.S.C. §103 ("§103")[1] prior art references, none of which were previously considered by the Examiners:

1) Peterson, Erik, *Building Applications with WebHub*, *Web Techniques* Magazine (July 1997) ("Peterson," Exh.1006);

2) Jensen, Cary, *Delphi in Depth*, Osborne/MacGraw-Hill (1996) ("Jensen," Exh.1007);

3) Papaj, Robert, *Oracle Databases on the Web Learn to Create Web Pages That Interface with Database Engines*, Coriolis Group Books (1997) ("Papaj," Exh.1008);

4) Surfas, Mark, *Running a Perfect Web Site with Windows*, Que Corp. (1996) ("Surfas," Exh.1009);

5) Chapman, Davis, *Building Internet Applications with Delphi 2*, Que Corp. (1996) ("Chapman," Exh.1010);

6) U.S. Patent No. 6,173,284, filed May 20, 1997, issued Jan. 9, 2001 ("Brown," Exh.1016);

---

1 The pre-AIA version of §103 is applicable to the '908 Patent.

7) Gladney, Henry, *Data Replicas in Distributed Information Services*, *ACM Transactions on Database Systems* 14:1 (March 1989) ("Gladney," Exh.1017);

8) U.S. Patent No. 5,548,753, filed to September 14, 1994, issued on August 20, 1996 ("Linstead," Exh.1018);

9) U.S. Patent No. 5,974,396, filed July 19, 1996, issued on October 26, 1999 ("Anderson," Exh.1019);

10) Texas Instruments C++ Object-Oriented Library User's Manual (1990) ("C++Library," Exh.1020).

2. <u>How the Construed Claims are Unpatentable under the Statutory Grounds Identified in 37 C.F.R. §42.204(b)(2) and Supporting Evidence Relied Upon to Support the Challenge.</u>

Pursuant to 37 C.F.R. § 42.204(b)(4) and § 42.204(b)(5), an explanation of how claims 1- 63, 77, 88, 99, and 100 of the '908 Patent are unpatentable under the statutory grounds identified above, including the identification of where each element of the claim is found in the prior art (see Exhibit Numbers), is provided in Sections V-VII below, including claim charts.

V. THERE IS A REASONABLE LIKELIHOOD THAT THE CLAIMS OF THE '908 PATENT ARE UNPATENTABLE

A. History of Databases and Database Updates

Organizing information into databases existed long before computers. Library card catalogs are analog databases. Exh.1014-¶40. With computers, the basic ideas behind databases did not change. Exh.1014-¶41.

Distributed database management systems (DDBMSs) operate across multiple computer systems. Exh.1014-¶72. By the 1980s, modifications at one database site could remotely update the same database hosted on other computers. *Id.* IBM developed systems synchronizing records on a personal computer with records on a remote server as part of the client/server paradigm in computing. Exh.1014-¶¶92-93.

Sending structured data via email messages became a standard no later than 1988. Exh.1014-¶97. By 1995, servers could send notifications of important database changes to clients. Exh.1014-¶99. Most database systems, including ORACLE, also provided functions for exporting query records into many different formats, including a compressed format. Exh.1004-¶39.

This technology advanced via the World Wide Web into Web-based database applications that were adaptations of these earlier client/server systems. Exh.1004-¶¶37,44; Exh.1014-¶¶100-101. These types of systems supported services such as Amazon.com and online real estate listings in the 1990s.

Exh.1004-¶¶48-57; Exh.1014-¶¶100-101. These services allowed users to use their personal computers to interact with and update a central database. Exh.1014-¶¶100-101.

B. Description of the Alleged Invention of the ’908 Patent

’908 Patent (Exh.1001) Fig.1 shows a web-updatable database system:

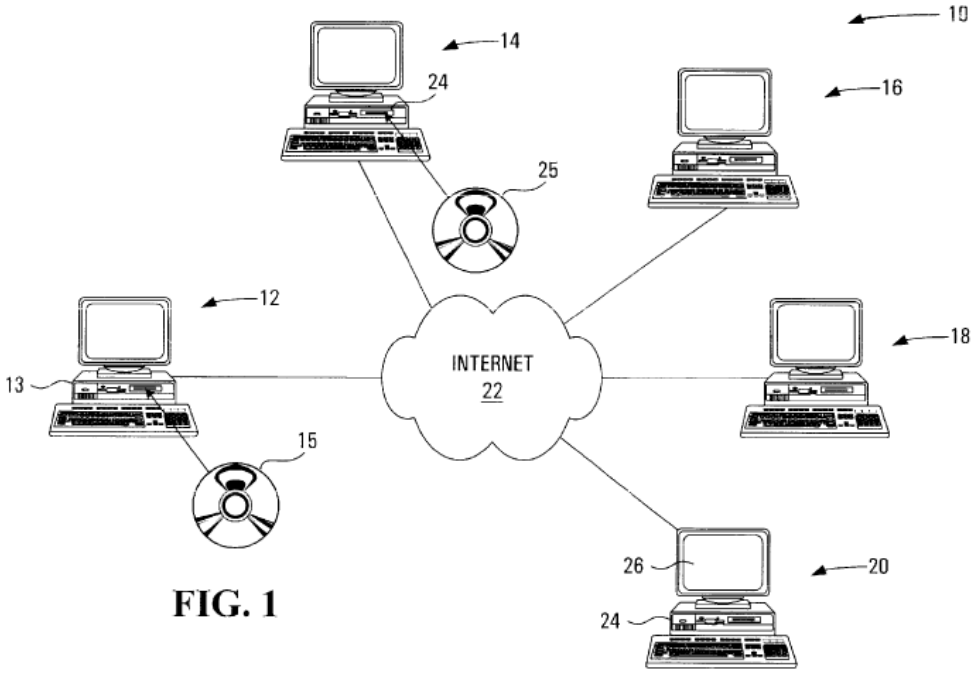


FIG. 1

Database system 12 and subscriber systems 14, 16, 18 and 20 enable users to communicate with the database server via internet 22. System 10 provides agents access to real-estate information they enter and accumulates, for a geographical region, for use by agents. (Exh.1001-8:29-38).

The prosecution history (Exh.1003) makes clear that the bulk of the claimed invention was well known in the prior art. Exh.1014-¶113. After rejections from

three Examiners (Exh.1014-¶¶114-119), claims were allowed based on the purported innovation of the "message server," the message including "a plurality of information records," and "compression." Exh.1014-¶¶120-122.

Claims 1-20 are virtually identical to corresponding claims 22-41 and 42-62. Claims 63, 77, 88 99 are essentially identical to Claim 1, and Claim 100 essentially Claim 1 + Claim 63. *See* Exh.1003, Exh.1004-¶¶117-118, Exh.1014-¶¶171-172. Therefore, Petitioner analyzes redundant claims by like groups denoted by {brackets}.

C. Declarations

Petitioner submits two 37 C.F.R. §1.68 declarations, Dr. Don Turnbull (Exh.1004) and Dr. Philip Greenspun (Exh.1014).

Dr. Turnbull (Ph.D., Information Studies, University of Toronto), focuses on software research and design for information systems, including the Web. Dr. Turnbull advises on software architecture for information systems, database implementation and design. He has been a professor at the University of Texas and a researcher at both Georgia Tech and the University of Toronto. He installed early Web servers and sites and investigated early Web technologies and databases related to eCommerce. He was also IBM's Lead Technical Architect building an Internet client/server platform for a multimedia client application combined with a database-driven Web site - the IBM-WorldBook Encyclopedia.

Dr. Greenspun (Ph.D.,E.E.,Comp.Sci, MIT) teaches Computer Science at the Massachusetts Institute of Technology (MIT). His thesis concerned engineering of large online Web-based communities using relational database management systems (RDBMS). In the 1990s, Dr. Greenspun developed web-based database applications for photo.net, which he founded, and numerous clients, including JobDirect, Hewlett-Packard, Oracle, and Siemens. He also wrote *Database Backed Web Sites*, published May 1997.

D. The Prior Art

1. Dr. Turnbull's (WebHub) References

a) Peterson (Exh.1006) describes functionality of the WebHub system. It demonstrates a web-based database apartment listing application.

b) Jensen (Exh.1007) describes the Delphi database development tool and the WebHub application framework. It discloses using the Delphi architecture for the functionality of WebHub, which simplifies traditional Web server data flow, serves Web pages, and exports to Email servers.

c) Papaj (Exh.1008) describes basic ORACLE database functionality and remote systems accessible through the Web and Email.

d) Surfas (Exh.1009) describes basic database functionality and discloses WebHub's functionality in creating websites with restricted subscribers and message notifications.

e) Chapman (Exh.1010) describes building and deploying database-driven Web applications with WebHub.

2. Dr. Greenspun's References

a) Brown (Exh.1016) discloses a system for notifying police officers of new or changed police records via email.

b) Gladney (Exh.1017) discloses a system transmitting a set of linked information records in a single message to achieve timely

dissemination of dated information and responses tailored for particular clients.

c) Linstead (Exh.1018) discloses a computerized DDBMS using email to send out machine-readable database updates to remote computer systems.

d) Anderson (Exh.1019) discloses a database analyzing purchasing information based on relationships in the database.

e) The C++Library (Exh.1020) discloses a comprehensive set of software tools for handling data encoded as bit-fields, keys and character strings.

VI. LEGAL DETERMINATIONS

A. Broadest Reasonable Construction

Claims receive the "broadest reasonable construction in light of the specification." 37 C.F.R. §42.204(b)(3), §42.100(b).

B. Person of Ordinary Skill in the Art

A POSITA had: (1) a Bachelor's degree in computer science or information systems (or related field) or four years education in computer science and software development; and (2) at least one year of relevant work experience building database-driven websites or client/server applications utilizing database management systems. (Exh.1004-¶31, Exh.1014-¶37).

C. Broadest Reasonable Interpretation of Claim Terms

All terms have their plain and ordinary meaning.

1. Message

Messages are any information sent or displayed to users, *e.g.*, via Internet protocols ("hypertext transfer protocol (HTTP)") or via email. Exh.1001, 1:39-40, 4:30, 14:29-45, claim 63.

2. Message Server

The message server "communicat[es] with the database manager for serving a message." Exh.1001, 2:3-4. Therefore, a POSITA would realize that "message server" has its plain and ordinary meaning as any server that sends a message to a user client.

3. Information Record

An information record is a data structure with information organized as a collection of fields. This term includes, but is not limited to, database records. Exh.1004-¶78.

4. Information Code

Information codes represent contents of field entries in records. Exh.1001, 11:9-10, Figures 7A-7B. An "information record" does not necessarily require the use of "information codes." Claim 1, for example, requires an information record,

and dependent claim 4 requires the contents of at least one field in the information record include an information code.

5. Common Field Entry

Common field entry is a standard term in the database arts. Relational databases "store[s] information in tables – rows and columns of data – and conduct[s] searches by using data in specified columns of one table to find additional data in another table. . . . In conducting searches, a relational database matches information from a field in one table with information in a corresponding field of another table." A POSITA would understand all relational databases can find records with matching field entries when conducting searches.

6. Compression

The '908 Patent broadly uses the term "compression." *See, e.g.*, Exh.1001, 2:50-51 ("compressed file may include an in image file and the message includes the compressed file"); 11:18-20 ("This [use of information codes] reduces memory storage requirements, effectively 'compressing' the information associated with each record."); 12:6-7 ("This [use of the key] effectively 'compresses' the building record."); 18:66-67 ("directs the processor circuit to produce a compressed exclusive 'zip' file"). Thus, to a POSITA compression is any activity that reduces the size of a set of data (or file).

7. Key

The ‘908 Patent illustrates a REGIONID key for the REGION table in FIG. 5. Ex.1001-005. A POSITA would understand “key” as something that identifies data or records.

8. Subscriber/Subscriber System

“Subscriber systems” are operable by subscribers (users) of the system. Exh.1001, 8:31-33. Thus, a POSITA would understand a “subscriber” or “subscriber system” as a synonym for the more conventional term “client,” but with the added implication that the client system receives updates. Each subscriber/subscriber system is, *e.g.*, a personal computer system, connected to the Internet and is capable of receiving a message from a server and, via local processing, presenting information from that message to the user.

9. Address/User Address

A “user address” is an address to which a message produced by the message server is to be transmitted. Exh.1001, 2:52-54. A POSITA would understand address can include, but is not restricted to, an email address.

VII. CLAIM-BY-CLAIM EXPLANATION OF GROUNDS OF UNPATENTABILITY

1) Peterson and Jensen. Peterson describes WebHub. Jensen describes the functionality of WebHub and Delphi. WebHub is a web-publishing framework

using the Delphi development language that creates dynamic websites and databases accessed through the Web. A POSITA necessarily views these references as obvious to combine because they ***were in fact combined*** by cross reference, and because together they disclose facets of the same complete WebHub system. Exh.1004-¶¶106,175; Exh.1006-004. *Optivus Technology, Inc. v. Ion Beam Applications S.A.*, 469 F.3d 978, 990–91, 80 U.S.P.Q.2d 1839, 1848–49 (Fed. Cir. 2006) (Where a prior-art reference cites a second prior-art reference in an obviousness analysis, a fact-finder can infer a motivation to combine); *Norian Corp. v. Stryker Corp*., 363 F.3d 1321, 1328, 70 U.S.P.Q.2d 1508 (Fed. Cir. 2004) (prior-art explicitly citing another prior-art reference provided sufficient evidence of motivation to combine).

2) Peterson, Jensen, and Papaj. Papaj describes ORACLE's database management system providing data organization for Web-accessible records. It would have been obvious to POSITAs to modify Peterson and Jensen's dynamic, web-based database applications to include database management and compression techniques from Papaj. Jensen discloses a database manager in the web-based database application, and Oracle as database manager. POSITAs understood that WebHub and Delphi must be compatible and combinable with Oracle. Papaj is an independent publication proving the underlying functionality of Oracle. Exh.1004-¶146.

3) Peterson, Jensen, Papaj, Surfas, and Chapman. Peterson also explicitly references Surfas and Chapman. Exh.1006-004. Surface and Chapman discuss the broad functionalities of WebHub and Delphi. A POSITA would also necessarily view these references as obvious to combine because they were in fact combined by cross reference, and together they disclose the multiple facets of the same complete WebHub system. Exh.1004-¶¶106,175. WebHub had already in fact combined the features these references describe.

4) Brown, Linstead and Gladney. Brown discloses a system notifying police officers of new database records matching search criteria at frequencies specified by a notification's requestor. The system relies on conventional RDBMSs accepting updates from remote clients. Notifications are sent via pager or electronic mail to a user or a "designated group of users." Like the '908 Patent, a principal goal of Brown is to make repeated Web page visits and search-criteria entry unnecessary.

Linstead discloses a system using email to support a DDBMS. The information in email notifications from Linstead's server to clients is not simply human-only-readable text; it includes a structured binary file.

Given intermittent Internet connectivity (a problem purportedly addressed by the '908 Patent) and/or system users primarily stationed at desktop computers having powerful local-processing capabilities, a POSITA would have ample reason

to combine Linstead, Brown, and Gladney to produce a system wherein users both receive and review alerts offline. With thousands of clients reading from the DBMS (*i.e.*, the use case described in the '908 Patent), it would be obvious to POSITAs to combine these references to allow users to: (a) register their specific interests (Brown), (b) browse a subset of the database with query results obtained from local memory (Gladney), (c) save the development cost of building communications software on both server and client by using email as a transport mechanism for structured data (Linstead), (d) use a table within the DBMS itself, and a daemon process, to record the need to send alerts and then send them (Linstead). 182. Therefore, it was obvious to POSITAs to modify Brown with Gladney and Linstead because doing so combines prior art elements (e.g., Internet web application of Brown with known distributed DBMS techniques of Gladney and Linstead) according to known methods to yield predictable results. Exh.1014-¶¶174-183

5) Brown, Linstead, Gladney, and Anderson. It would have been obvious to a POSITA to use the teachings of Brown, Linstead, and Gladney in accordance with standard data modeling techniques known to POSITAs and exemplified in Anderson. Exh.1014-¶¶163,249,250.

6) Brown, Linstead, Gladney, Anderson, and C++Library. A POSITA would know to combine the above with bit vector processing, keys, and character

strings known to any POSITA as evidenced by the standard teachings of the C++Library and popular programming language. Exh.1014-¶¶158-161. POSITAs would combine conventional RDBMS references like Brown, Gladney, Linstead with the bit-vector processing disclosed in C++Library to facilitate flexible searches or to cope with a shortage of disk space. Exh.1014-¶¶278-286.

A. Ground 1

Peterson and Jensen render claims 1{22,42,62}, 2{23,43}, 21, 63{77,88,99}, and 100 obvious under §103 (*see also* Exh.1004-00111-00121):

1.	A **remotely updatable database system** comprising: **{22,42,62,100}**
	"The [WebHub] application [CityQ] supports several tables, pulls HTML from a SQL **database** server, and allows simultaneous, **remote Web-site updates** by multiple users." Exh.1006-003.
1(a)	a **user interface** for **communicating with at least one subscriber system** to **receive user input** from a user at said at least one subscriber system; **{22,42,62,100}**
	Peterson's WebHub application, CityQ, provides a user interface for communicating with subscribers. Figure 2 of Peterson illustrates how CityQ utilizes a **user interface** to **communicate** with a Web user's computer (**a subscriber system**). **User input** includes checking or unchecking options; selecting sorting criteria; **inputting** map coordinates, zip codes, prices or number of bedrooms; or clicking the "Search" button. Exh.1004-¶124. "The **user** first sees the HTML form shown in Figure 2, **enters the search criteria**, and presses the Search button at the top of the form." Exh.1006-002.

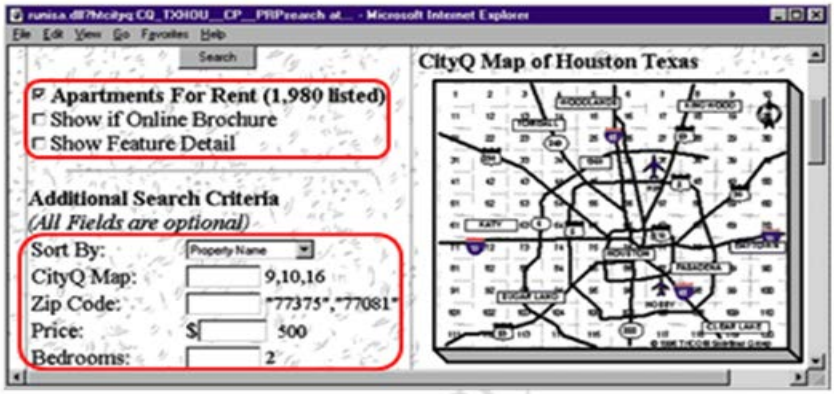
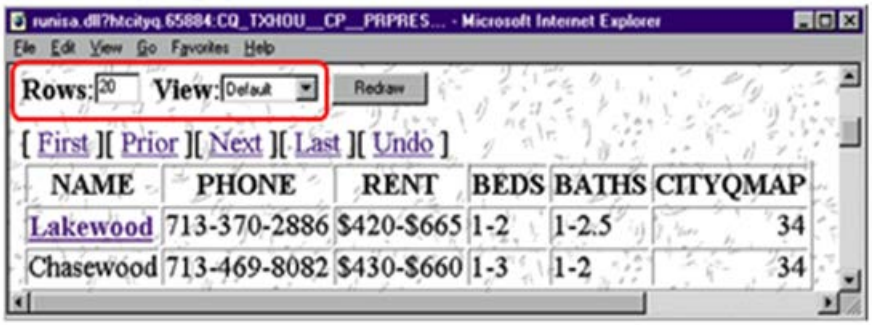
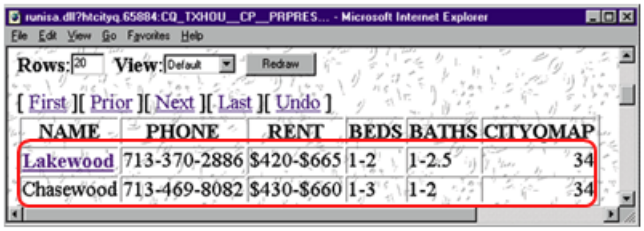
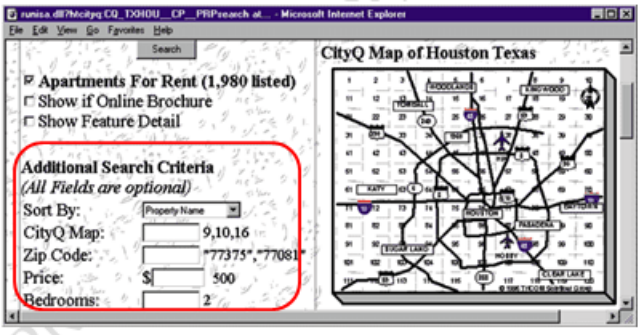
	 Figure 2 Exh.1006-006. "The **user** controls the **number of records** or rows shown at a time, can change views to see additional information, and can use First, Prev, Next, and Last buttons to navigate the result set." Exh.1006-002.  Figure 3 Exh.1006-007.
1(b)	a database of **information records** and **user records**, said information records having respective **pluralities of fields** **{22,42,62,100}**
	"This Web app makes a **database** of about 2000 apartments in the Houston area searchable. People looking for an apartment can **search by location, price range, number of bedrooms, zip code, name, and so on**." Exh.1006-002. Each of the apartment records in the database is an **information record**. The information records have a **plurality of** searchable **fields** including location, price range, number of bedrooms, and zip code. Exh.1004-¶126.

	Figure 3 of Peterson illustrates a Web interface showing **a database of information records**. Exh.1004-¶127.  Figure 3 Exh.1006-007. For "**user** record" see 1(c) below.
1(c)	and wherein each of said information records is **linked to at least one user record**; **{22,42,62,100}**
	The "result page" shown in Figure 3 of Peterson displays resulting sets of **information records linked to the user record** by the user's input into the user interface shown in Figure 2. Exh.1004-¶128.  Figure 2 Exh.1006-006. Figure 3 shows more than one record (**records**, *e.g.*, Lakewood, Chasewood**)** linked to the **user record** with **pluralities of fields** (shown as "NAME", "PHONE", "RENT", "BEDS", "BATHS" and "CITYQMAP") in the form of a **user record** of search results in the CityQ WebHub system. Exh.1004-¶128.

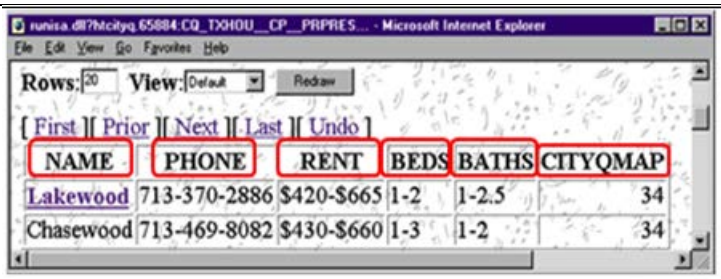


Figure 3

Exh.1006-007.

Jensen describes linking and association via WebHub's Session ID.Exh.1004-¶129. *See* Exh.1007-0035 ("The Session ID basically provides a key into the "database" of **saved state information**. In other words, there will be **saved-state data** for each surfer, **keyed by Session ID**. Once the ID has been assigned, it is used to create and maintain a **set of data** for the surfer.").

1(d)	a **database manager** in **communication with said user interface**, for **controlling** said database such that each information record is associated with at least one user, **{22,42,62,100}**

Figure 17-2 in Jensen illustrates **database manager** components **in communication with a user interface** (the Web browser) via WebHub Central. WebHub Central manages the database to **associate the information records with at least one user** (custom.exe built with WebHub components).Exh.1004-¶130.

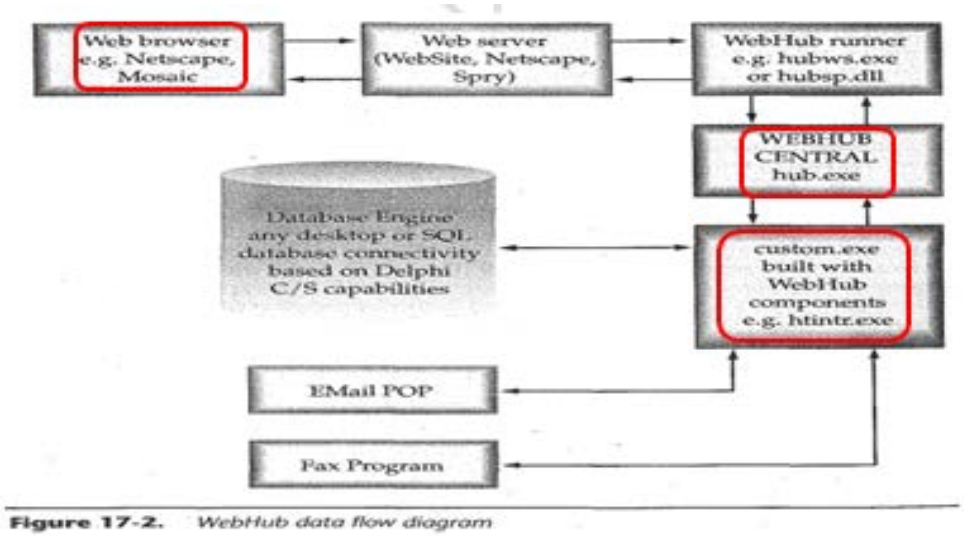
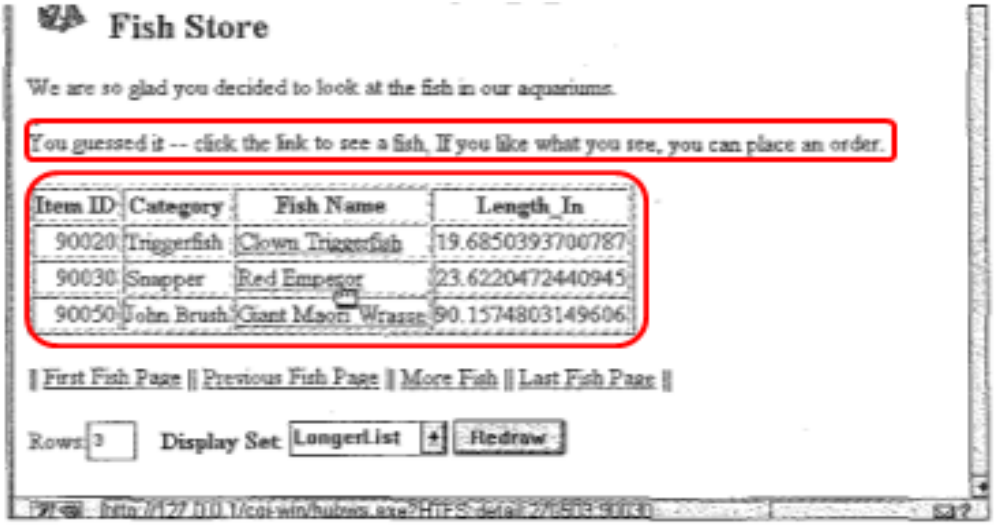


Figure 17-2. *WebHub data flow diagram*

Exh.1007-0017.

	Also see 1(c) above for references disclosing "information records linked to at least one user record."
1(e)	**{22,42,62,100}** and for **amending said information records in response to user input** received at said user interface from said at least one subscriber system; and
	A POSITA understood database systems provided the ability to **create, read, update and delete**, therefore **amend an information record**. Thus, to a POSITA, the several WebHub references herein describing "**updating**" information records also disclose "**amend an information record**." Exh.1004-¶131. User input as shown in Jensen Figures 20-3, 20-4 and 20-5 below allows the user to communicate with the database manager via the user interface to **amend** her user **record** with the potential fish order, as a subscriber or user to the database manager of the fish database. Similarly, when the Fish Store adds a new fish available for purchase or changes the price on a particular fish, that **information record** is added or **amended in response to user input**. Exh.1004-¶132.  **20-3.** *The lookfish page from the Fish Store* Exh.1007-0075.

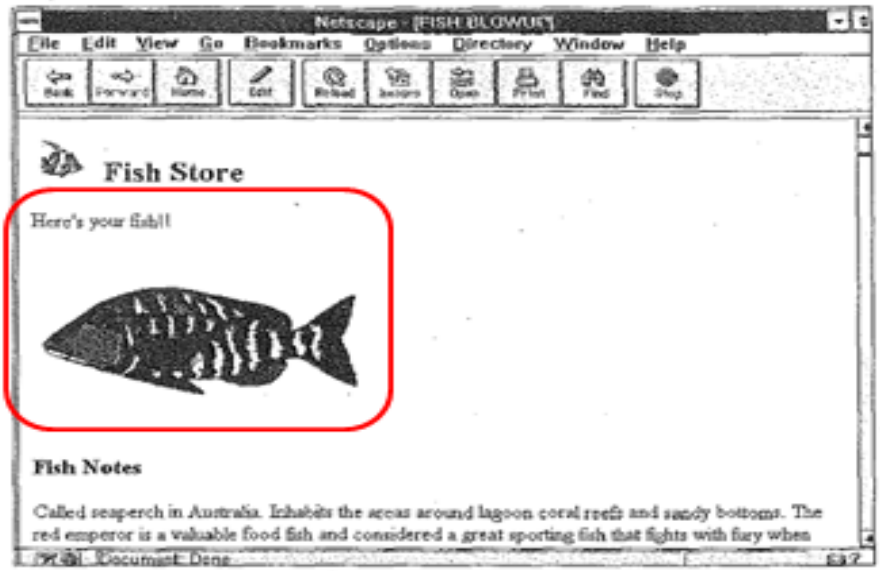


Figure 20-4. *Detail about the Red Emperor fish*

Exh.1007-0076.

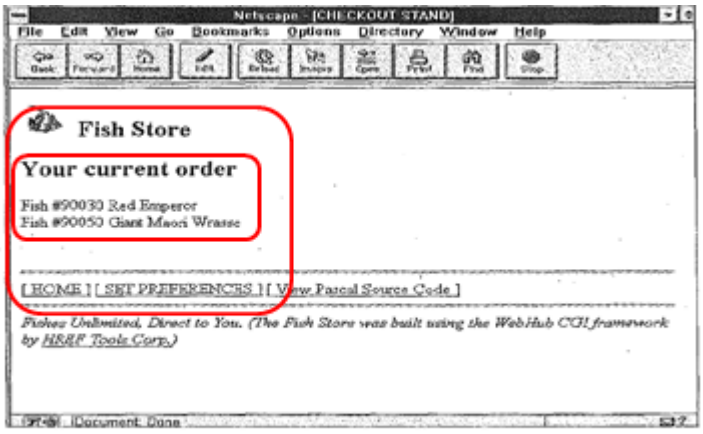


Figure 20-5. *The checkout counter, which lists the selected items*

Exh.1007-0076.

Jensen figure 20-3 (Exh.1007-0075), the **user can request** how many rows to show (in the example, it is 3). If user changes that, they are **amending the query** to show more rows. See also Figure 3 of Peterson (Exh.1006-007), where the user can amend the query to see a certain number of rows ("20") per page. Exh.1004-¶133.

1(f)	a **message server** in communication with said database manager **for serving a message including at least one record received from said database manager to said at least one user** associated with said information record, **{22,42,62,100}**

Figure 1 of Peterson (Exh.1006-005) shows "the flow of information using WebHub" featuring both servers and databases. The "Web Server" in this diagram is a **message server** because it **communicates with the database manager** shown in the diagram as "Databases." As shown below, **records can be received from the database manager**. Exh.1004-¶134.

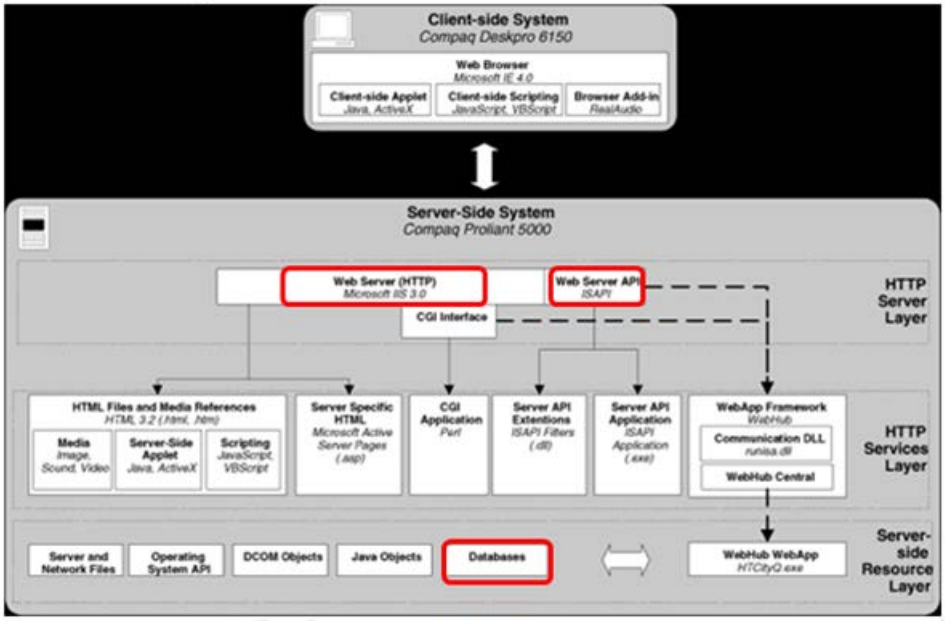


Figure 1

Exh.1006-005.

Figure 3 of Peterson further shows records received from **communication with said database manager**, and the **record being associated** with and sent to the user. Exh.1004-¶135.

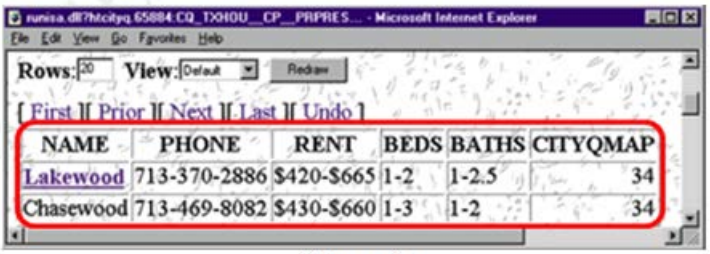
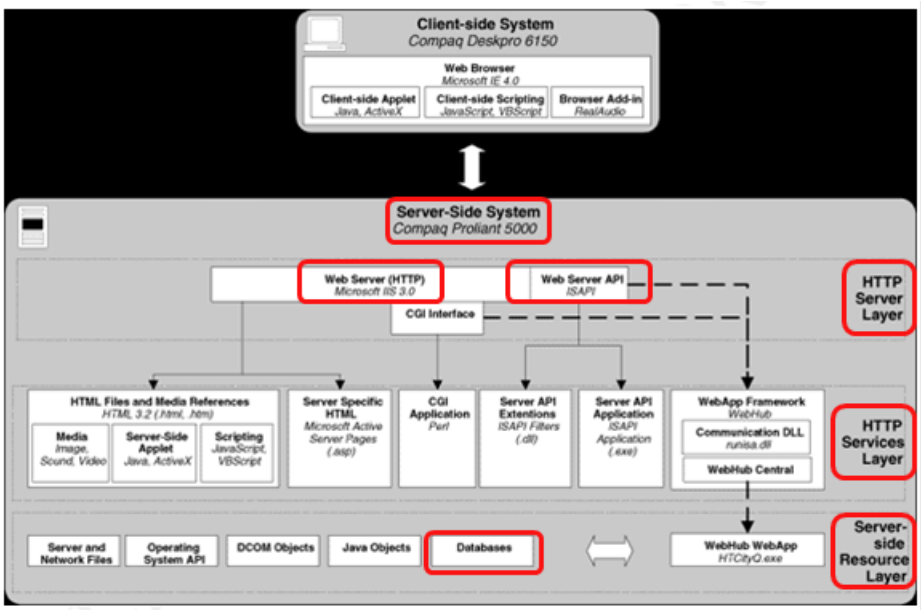


Figure 3

Exh.1006-007.

CityQ serves to the user the content of the information records resulting from the user's database query as **messages** displayed in the CityQ Web interface (*e.g.*, Figure 3 of Peterson) or in an email for message distribution (*e.g.*, "Email Pop" in Jensen Figure 17-2, Exh.1007-0017). Exh.1004-¶135.

1(g)	said message server including **a processor and memory for storing program codes** readable by said processor **to direct said processor to communicate with said database manager** **{22,42,62,100}**
	Figure 1 of Peterson shows the "Server layer" operating on a Compaq Proliant 5000 server would include a processor and memory for **storing program codes**… A **database manager** is shown in the "Resource Layer." The system must have the ability **to direct said processors to communicate with said database manager** by requesting and passing data in a data flow. Exh.1004-¶138.  Figure 1 Exh.1006-005.
1(h)	to obtain for **inclusion in said message a plurality of information records having at least one common field entry.** **{22,42,62,100}**

	The system passes data through a data flow as described above in section 1(g). This could include a query based on a user's interest that can be **information record** data suitable for inclusion in (said) **message**. Exh.1004-¶139. "The Web app dynamically generates and executes an SQL query to the database. The result is returned to the Web app, **formatted into an HTML table**, and passed first to the HTTP server and then **back to the user**. … This search resulted in two apartments **matching the user's criteria**." Exh.1006-002. A **common field entry** would be a record matching a user request from the database manager. The **common field entry** in Peterson Figure 3; *i.e.*,"CITYQMAP" field showing two information fields matching "34." Like any web app, WebHub "must be able to access server-side resources just as easily as other programming environments. It must be able to access files, make operating-system calls, **query databases**, use mail protocols, or access any other resource available to development tools such as Delphi, C++, or Visual Basic." Exh.1006-001. WebHub therefore accesses server-side resources including the database manager (database server) and can "query databases" for **common field entries**. Exh.1004-¶141.
2.	... claim 1 wherein said message server includes a processor and memory for storing program codes readable by said processor to direct said processor to communicate with said database manager to obtain for inclusion in said message **at least one** information record having at least one common field entry. **{23,43}**
	See Claim 1 above for references reciting majority of these claim limitations. Peterson Figure 3 (Exh.1006-007) shows the two apartments found when the user input search criteria in the Peterson Figure 2 interface. Exh.1004-¶143.
21.	... claim 1 wherein said user interface includes a **web interface**.
	Peterson Figure 3 (Exh.1006-007) illustrates a **Web interface** showing a database of information records. Exh.1004-¶144.
63.	An apparatus for receiving **database record information** from a database containing a **plurality of database records**, the apparatus comprising: a) a message receiver for receiving **a message from said database**, said

	message including at least one file from said database; b) transfer memory for storing said at least one file; c) an **output device for presenting information to a user**; and d) a **processor circuit in communication with** said message receiver, said transfer memory and said output device, and **processor memory programmed with processor-readable codes for directing said processor** to direct said output device to present to the user a list of files stored in said local memory, the files included in said list having **at least one predefined characteristic**; and to direct said output device to present to the user at least some of the contents of at least one file having said at least one predefined characteristic. **{77,88,99,100}**
	See Claim 1 above for references reciting majority of these claim limitations. See Exh.1003. Peterson Figure 3 (Exh.1006-007) discloses display of information in the CityQ Web interface to users. Exh.1004-¶95. Transfer memory is inherent to computers, and replacing existing files in said transfer memory with new files received from said database is inherent to any distributed database update. Exh.1004-¶144. A POSITA would understand a file is a basic unit of storage of information and synonymous with information record. Similarly, field contents are synonymous with predefined characteristics. Exh.1004-¶144. *See*, Exh.1001-0039, 11:4-7 ("information codes representing a predefined character").
100.	Claim 100 is an obvious and verbatim combination of Claims 1 and 63.
	See Claims 1 and 63 above for references reciting the majority of the claim limitations. Moreover, central servers are typical in database applications, and in any event not novel but completely obvious as of the priority date. Exh.1004-¶146.

Peterson discloses a Web server (WebHub) supporting databases. Exh.1004-¶120. Peterson discloses a user interface that interacts with users. Exh.1004-¶124. This interface can include tables a user can format and control. Exh.1004-¶193. Peterson discloses a database (Web-app listing of Houston apartments) containing information and user records generated in response to and indicative of users searching for apartments available for rent. Exh.1004-¶¶126-127. Peterson discloses a server communicating with the database manager and sending messages to the user. Exh.1004-¶120. To create the display of Figure 3 from Figure 2, Peterson's server includes processor and memory for storing program codes. Exh.1004-¶138. Peterson shows the Web-app showing search results in a database display in an HTML table, thus including at least one information record in the message shown to the user. Exh.1004-¶139-140. These records would have a common field based on user search: for instance, if a search was done for a certain price range the common field of the displayed records is price. Exh.1004-¶141.

Jensen (Figure 17-2) discloses components of a database manager communicating with a Web-browser user interface. Exh.1004-¶108. This necessarily includes the ability to create, read, update, or delete, *i.e.*, "amend," records. Exh.1004-¶131. An example is Jensen's "fish store." Exh.1004-¶109. Jensen describes Delphi as a programming language and development environment in which HREF software's WebHub is written to provide an easy framework for

database-driven Web sites. Exh.1004-¶121. Peterson then describes an example of a WebHub-powered Web application, his CityQ database-driven Web for searching a remote database of available real estate with an interactive user interface. *Id*.

It would have been obvious to one of ordinary skill in the art to "combine" the teachings of Peterson with the teachings of Jensen because Peterson cites Jensen itself and additionally both references discuss the same product (WebHub). Exh.1004--¶120. The disclosures of Peterson and Jensen were already combined because both references discuss the functions and application of WebHub and Delphi. Peterson and Jensen merely focus on different aspects (and examples) of WebHub and Delphi. *Id*.

B. Ground 2

Peterson, Jensen, and Papaj render claims 3-7{24-28,44-48} and 14-20{35-41,55-61} obvious under §103 (*see also* Exh.1004-00122--00130):

3.	... claim 1 wherein said database manager **stores the contents** of at least some of said information records **in a compressed format**. **{24,44}**

Listing 5.1 A sample function to classify a customer type.

```
create or replace function get_customer_type (cust_id varchar2)
return boolean is
          average_price   real;
          declare c1 is

          select avg(item_price)
          from
          line_item, item, order
          where
          order.customer_id   = :cust_id
          and
          order.customer_id   = line_item.customer_id
          and
          line_item.item_nbr  = item.item_nbr;
BEGIN
```

Exh.1008-0023.

An example of one type of "compression,' Papaj 5.1 above discloses "cust_id" in the ORDER table, which is a key and an information code, instead of the customer name. Exh.1004-¶148.

4. ... claim 3 wherein said **contents of at least one field** of said information record **includes at least one information code**.
{25,45}

The value of "34" found in the CITYQMAP field in Figure 3 of Peterson. Exh.1004-¶150.

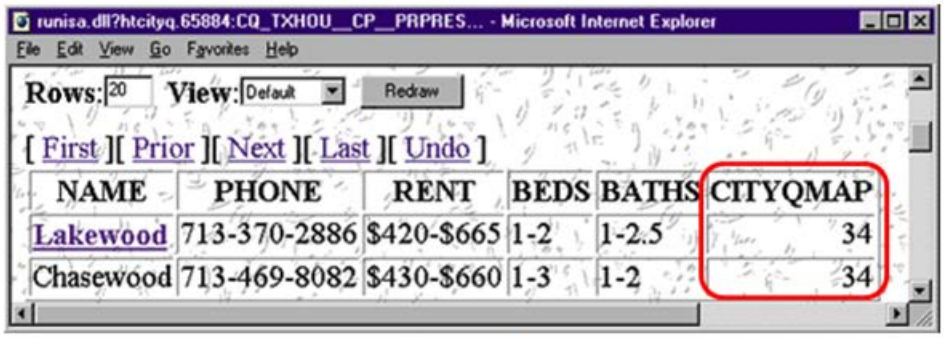


Figure 3

Exh.1006-007.

The **information code** "34" represents a predefined value **contents of one field** which corresponds to information regarding the location of the apartment on the CityQ Map illustrated in Figure 2 of Peterson. Exh.1004-¶150.

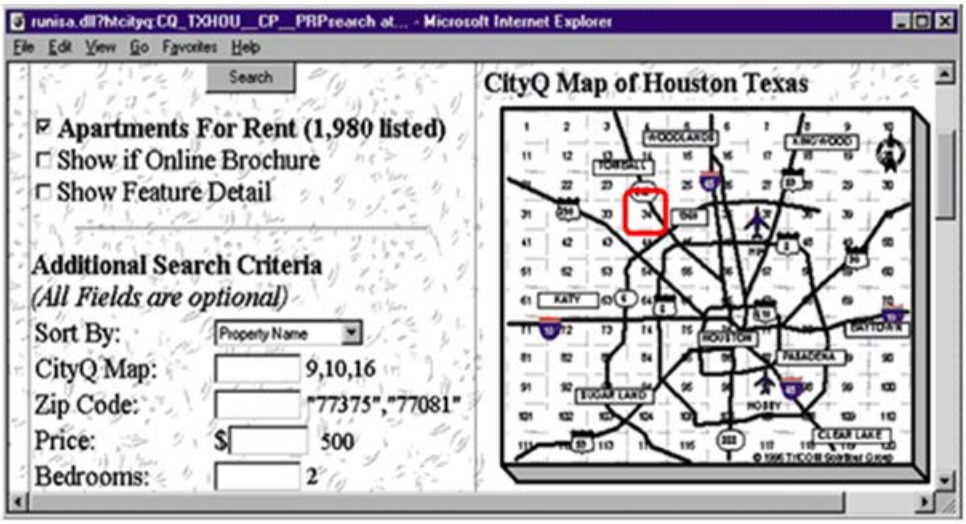


Figure 2

Exh.1006-006.

Jensen Figure 19-5 below also shows information codes, the "Code" field for a record ID, and "LP" instead of "long playing." Exh.1004-¶151.

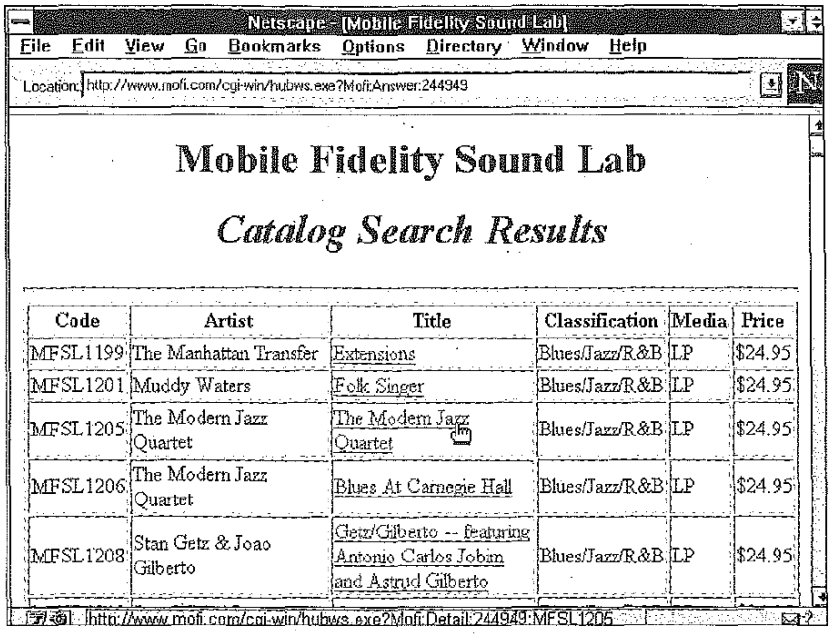
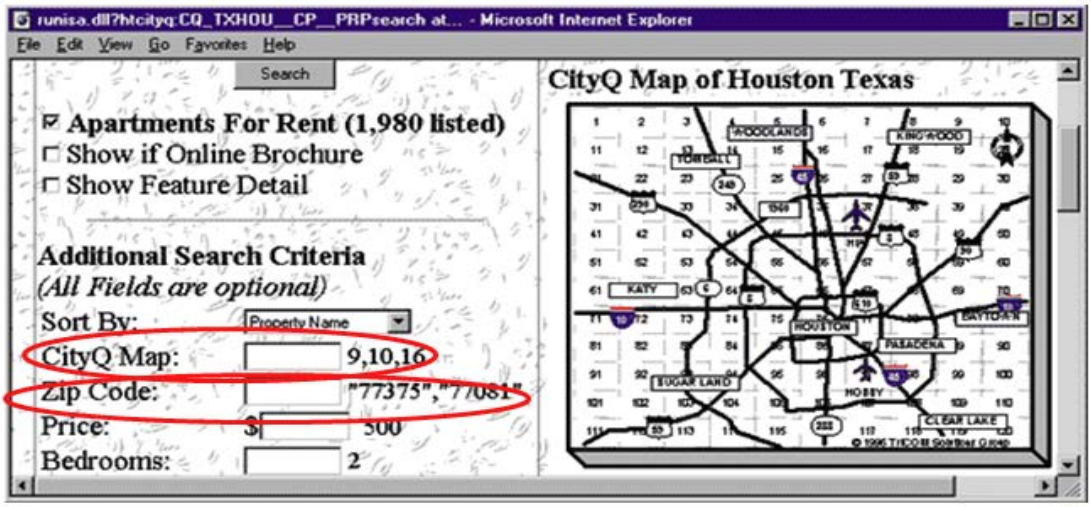
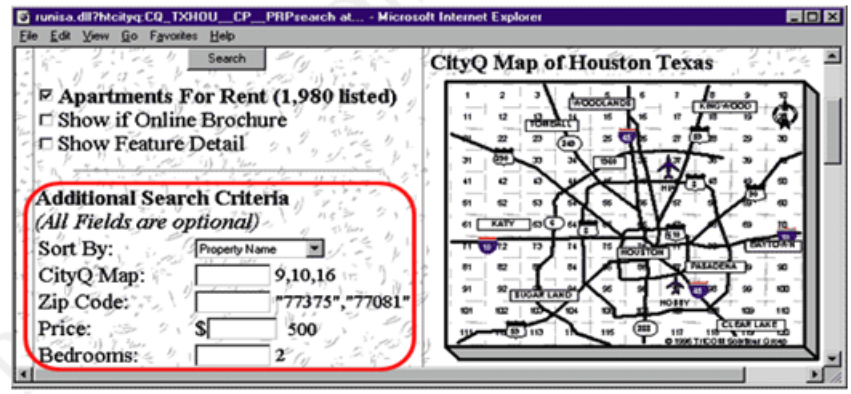


Figure 19-5. *HTML table displaying the result of a search*

Exh.1007-0037.

	Jensen also discloses information codes when referring to using a custID key. Exh.1004-¶151. custID=123987 Exh.1007-0045.
5.	... claim 4 wherein said at least one information code **represents at least one of a plurality of predefined contents** of said field. **{26,46}**
	Information code "34" in the CITYQMAP field of Peterson's Figure 3 is an information code that represents a predefined item. Peterson Figure 2 explicitly provides an example to the user that a query can encompass multiple codes in a single field ("9,10,16"). Exh.1004-¶153. Peterson Figure 2 shows several database fields in a search (query) interface that allows selection or input of an information code matching the **predefined content of a field** in the database manager. Exh.1004-¶153.  Peterson Figure 2 Exh.1006-006. In Peterson, an **information code** is the **predefined** set of 5-digit strings representing the zip codes in the Houston area that would be in the database manager's field for zip codes and a **plurality of field contents** may match a

	zip code string. Exh.1004-¶154.
6.	... claim 4 further including **memory for storing a key** identifying said predefined contents of each of said fields represented by said at least one information code. **{27,47}**
	In Peterson, a **key** used to **identify** (and therefore distinguish) a piece of information such as the "Property Name" which would be a string identifying each apartment complex or apartment building in the CityQ database, or the grid of the CityQ Map to decode the value of "34." As shown in Figure 2, when a user inputs a choice of a Property Name to "Sort By" and selects or inputs other search criteria into the CityQ user interface, a **plurality of pieces of information** are displayed that match the key(s) selected. ("Lakewood" and "Chasewood" in Figure 3). Exh.1004-¶155. "The Session ID basically provides a key into the "database" of **saved state information**. In other words, there will be **saved-state data** for each surfer, **keyed by Session ID**. Once the ID has been assigned, it is used to create and maintain a **set of data** for the surfer." Exh.1007-0035. Instead of using the customer name, Papaj discloses the use of CUSTOMER_ID in the ORDER table (*i.e.*, a key and an information code). Exh.1004-¶157; Exh.1008-0023.
7.	... claim 6 wherein said server memory includes **program codes for directing said server processor to produce a composite file** including said key and at least one of said information records. **{28,48)**
	Peterson Figure 3: the results of a database search are produced from the Web server and shown as an HTML **composite file** (*i.e.*, consisting of more than one element). Exh.1004-¶158.
14.	... claim 4 wherein said at least one information code represents a **quantitative measure of a predefined physical quantity**. **{35,55}**
	Figure 2 of Peterson illustrates how many of the **information codes** in the search (query) interface **represent a quantitative measure of a predefined**

	physical quantity such as a "CityQ Map" **quantity**, a "Zip Code" **number,** representing a **quantified physical location, and number** of bedrooms as well as the "Price," which are each a **quantitative measure**. Exh.1004-¶160.  Peterson Figure 2 Exh.1006-006.
15.	... claim 14 further including **memory for storing a key identifying said predefined physical quantity**. **{36,56}**
	See Claim 6 for references reciting memory for storing a key identifying said predefined contents of each of said fields represented by said at least one information code, and predefined physical quantity. Exh.1004-¶161.
16.	… claim 15 wherein said key includes a **character string** associated with said field. **{37,57}**
	Peterson shows in the WebHub application CityQ example where a key such as "Property Name" or "Bedrooms" can contain **a plurality of character strings**, such as the name of a property or location. Exh.1004-¶162. In Figure 3 of Peterson, **character strings** are database **key fields**, for example the NAME field with **character strings** "Lakewood" and "Chasewood." Exh.1004-¶162.

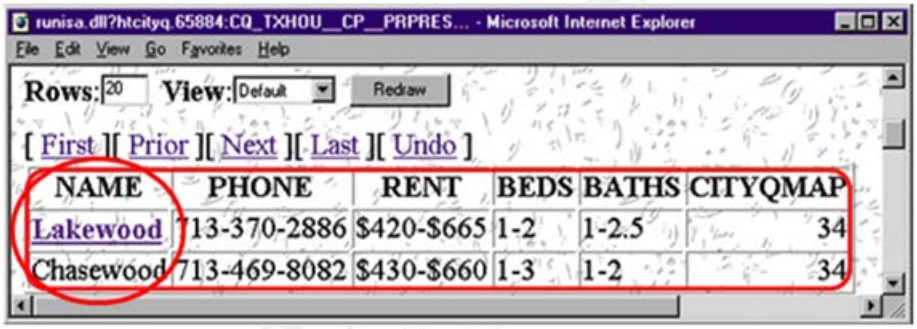


Figure 3

Exh.1006-007.

17.	... claim 4 wherein said at least one information code identifies whether or not any of **a plurality of pieces of information are to be associated with** said record. **{38,58}**
	In Figure 3 of Peterson, the "CITYQMAP" field shows two (**a plurality of**) records with information codes matching a "34" information code, along with other associated pieces of information including the character strings "Lakewood" or "Chasewood" in the NAME, the PHONE field ("713-370-2886" or "713-469-8082"). Exh.1004-¶163.

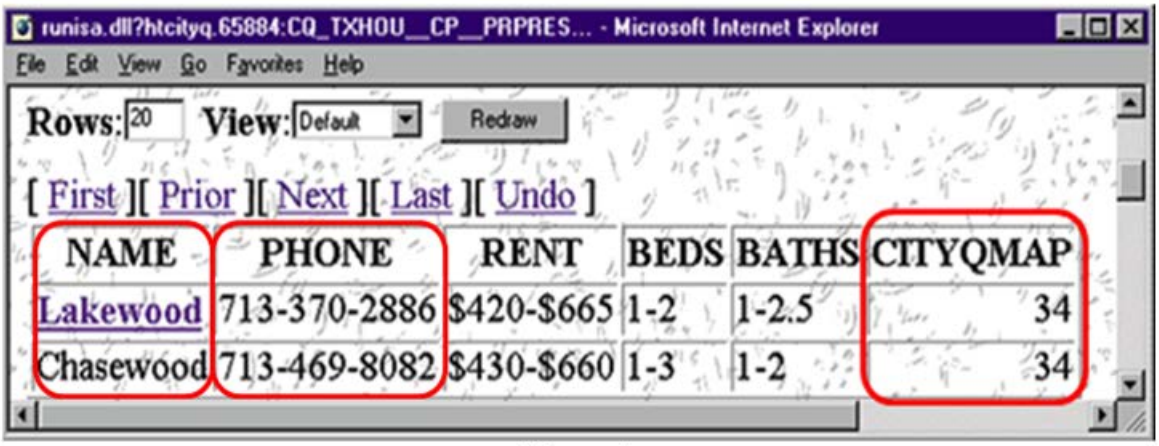
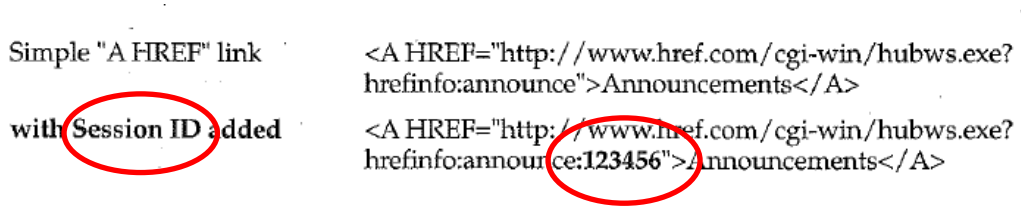


Figure 3

Exh.1006-007.

In this example, the **information code** "34" **identifies that the plurality of pieces of information** for the two apartment complexes that are associated with said record, and are therefore displayed. Exh.1004-¶164.

Another example of **information codes** in Peterson, are the pre-defined set

	of 5-digit strings representing the zip codes in the Houston area. **Any of a plurality of pieces of information (that) are to be associated with said record** are the other fields such as the CityQ Map **information code** or the price **information code in said record.** Exh.1004-¶165.
18.	... claim 17 wherein said at least one information code includes **a decimal number representing a binary number having individual bits representing the inclusion or exclusion of a corresponding predefined piece of information**. **{39,59}**
	Exh.1007-0038:  This Session ID decimal number "123456" lets WebHub identify the user and only include the user input (a query, or **set of data for the surfer**) as the **predefined pieces of information corresponding** to the user Session ID "123456". Thus the "123456" is an information code **that includes a decimal number representing a binary number having individual bits representing the inclusion or exclusion of a corresponding predefined piece of information**. The query results are included (or excluded) based on the information code "123456" matching (or not matching) the corresponding predefined piece of information. Exh.1004-¶169. "Once the ID has been assigned, it is used to create and maintain **a set of data for the surfer**. (WebHub does this with a TWebSession component… There are ways to control whether the state data is kept only in RAM, saved to **binary disk files**, or saved in a database, for example.)" Exh.1007-0035.This shows the decimal Session ID can also be saved as a **binary number.** Exh.1004-¶170. In Peterson, one example of **at least one information code** is the "price" of apartments in the CityQ database. Prices are commonly represented as **a decimal number information codes**, which would allow the database, when searched using the price to match or not match based on the particular price queried. Exh.1004-¶171.

	At least one information code is the "CityQ Map" decimal number, which would allow the database, when searched using the "CityQ Map" **information code, to include or exclude results based on the information code matching the corresponding predefined piece of information**. Exh.1004-¶171. Papaj also discloses using bit fields. Exh.1004-¶172; *see* Exh.1008-0023. ("Now that the boolean value of **big_spender** and **ordinary** customers has been set, theWebServer application can test this value to determine the appropriate images and frames to place on the dynamic Web page.")
19.	... claim 18 further including **a key** identifying said plurality of pieces of information. **{40,60}**
	See Claim 6 for references reciting memory for storing a key identifying said predefined contents of each of said fields represented by said at least one information code. Exh.1004-¶173.
20.	... claim 19 wherein said **key includes a plurality of character strings** associated with said field. **{41,61}**
	See Claim 16 for references reciting a key including a plurality of character strings. Exh.1004-¶174.

In addition to the above disclosures, adding Papaj to Peterson and Jensen renders these dependent claims obvious.

Papaj discloses the use of keys and information codes, which illustrates compression and programs within Oracle that create compressed formats. Exh.1004-¶148. Papaj also discloses using bit fields. Exh.1004-¶173.

Peterson discloses using pre-defined codes. Exh.1004-¶¶151-52. Peterson uses these codes to represent certain fields, such as apartment location on the CityQ map. Exh.1004-¶¶153-55. Peterson also discloses the use of a key to identify the value of an information code, for example the Session ID key. Exh.1004-¶¶156-58. A key in any modern database can contain a plurality of character strings associated with a field that the key identifies within the database. Exh.1004-¶¶162-63. Peterson discloses that those codes represent a quantitative measure (*e.g.*, geographical location or price). Exh.1004-¶161. Peterson also discloses an HTML composite (*i.e.*, consisting of more than one element) file. Exh.1004-¶¶159-160. Peterson discloses that the information records displayed in the HTML file can match records to certain pieces of information, for instance common locations or price range. Exh.1004-¶¶164-167.

Jensen also discloses the Session ID number is a decimal number representing a binary number. Exh.1004-¶¶168-75. This number directly corresponds to a piece of information, such as apartment price. Exh.1004-¶¶168-75.

It would have been obvious to a POSITA to modify Peterson and Jensen to include database management and compression techniques from Papaj. Exh.1004-¶146. Jensen discloses using Oracle as a database manager. *Id.* As such, a POSITA would understand that WebHub and Delphi are compatible and combinable with

Oracle. *Id.* Thus, it would have been obvious to one of ordinary skill in the art to use Oracle, which includes the functionalities disclosed in Papaj, as the database manager in a WebHub and Delphi application, as taught in Jensen, to yield the predictable results yielded by the system to which the claims of the '908 Patent are directed. *Id.*

C. Ground 3

Peterson, Jensen, Papaj, Surfas, and Chapman render claims 8-13{29-34,49-54} obvious under §103 (*see also* Exh.1004-00131--00135):

8.	... claim 7 wherein said server memory includes program codes for directing said server processor to produce **a compressed file** including said key and at least one of said information records. **{29,49}**
	See Claims 3 and 7 above for references reciting the majority of the claim limitations. Exh.1004-¶176. As discussed, instead of using the customer name, Papaj discloses compression via the use of CUSTOMER_ID in the ORDER table (*i.e.*, a key and an information code). Exh.1004-¶178; Exh.1008-0023. "A capability similar to MIME types is encoding, in which the server can mark compressed documents as being encoded in a specified format. Browsers that support automatic decoding can then use the encoding type information to automatically decode the file when it is received. You indicate encoding types with the AddEncoding directive." Exh.1009-0020. A Web server, which uses HTTP, includes a server processor and server memory, can serve .gzip files, compressed files, upon request or by default depending on the configuration. Exh.1004-¶181. "Content - Encoding enables HTTP to transport data with representations other than ASCII, coordinating the actions between server and client. For example, Content Encoding: gzip enables the server to transmit a gzip compressed file to the

	client and have the client decompress the file." Exh.1010-00110. The subscriber system user's Web browser can de-compress these compressed files for display in the Web browser user interface when downloaded. Exh.1004-¶177. *See also* Exh.1009-0020.
9.	... claim 8 wherein said **compressed file includes an image file**. **{30,50}**
	A POSITA would understand that almost all image files are already compressed, as noted in Surfas. Exh.1004-¶182. "JPEG can handle so many colors in a relatively small file because it **compresses the image data**. You can control how big or small the **image file** ultimately is by adjusting the parameters of the **compression**. A highly **compressed file** can be very small, but the quality of the **image** on-screen will suffer for it." Exh.1009-0095. "The **Image data type** is the one most commonly used to **transfer binary files** such as **images** and applications (and .ZIP files, Word .DOCs, and so on)." Exh.1010-007. "In the early days, the data was all text-based. …. These days, the data that you send is more likely to be a word processing document a **graphic image**, or a sound or video clip." Exh.1010-0068. Figure 20-4a demonstrates a Web page interface for a subscriber system that **includes an image file**, which can be **compressed** by the Web server. Exh.1004-¶185.

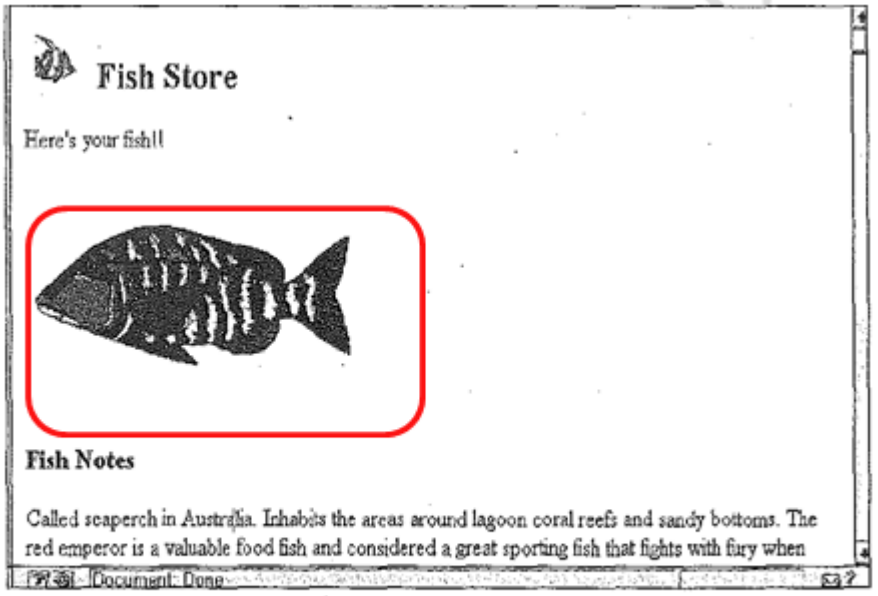


Figure 20-4. *Detail about the Red Emperor fish*

Exh.1007-0076.

10.	... claim 9 wherein **said message includes said compressed file**. **{31,51}**
	See Claim 8 for reference reciting program codes which direct the server processor to produce a compressed file. Exh.1004-¶184. All popular Web servers supported file compression at the time. Some Web server protocols could natively compress data for transmission to a receiver. One example is the SSL (Secure Sockets Layer) protocol "fragments messages into manageable blocks. SSL can compress the data before adding a signature and encryption. When this compression process is complete, SSL transmits the message." Exh.1004-¶186. *See* Exh.1010-00139.
11.	... claim 10 wherein said **user records include a user address field** for storing a user address to which a message produced by said message server is to be transmitted. **{32,52}**
	Peterson describes how any "Web app" would interact with users by tracking them throughout the Web application use session to transmit a message (in this case the graphic includes the user's name). Exh.1004-¶188. "Say you wanted to create a customized GIF that superimposes the surfer's name, in a

	user-specified color and font, as a border around an action photo of an NBA player. After the Submit button is pressed, the server would pass the command to a Web app, which would dynamically create the content (based on state or session variables), send it back to the HTTP server, and finally return it to the user." Exh.1006-001. To transmit the message, Peterson's example must include **a user field** with a **user address stored** in it. Exh.1004-¶189. Peterson Figures 2 and 3 show messages that are **produced by said message server to be transmitted** from the message server into the Web browser on the user's subscriber system. The user address for this message can be the WebHub SessionID or the user's subscriber system IP address. Exh.1004-¶190.
12.	... claim 11 wherein said memory includes program codes for directing said processor to direct said database manager **to scan said user records to obtain a distribution list of user records** having the contents of at least one field associated with the contents of a predesignated field in at least one of said information records. **{33,53}**
	A POSITA would understand that any database manager provides the functionality to search, query or **scan** a set of **(user) records** for content, including **the contents in the fields of the information records** stored in the database, with the query results being a **list of user records**. Exh.1004-¶193. In Figures 2 and 3, Peterson shows the WebHub application CityQ provides a user an interface to input a search into the CityQ database to scan records matching **at least one field associated with the contents of a predesignated field**, such as "Zip Code," "Price" or "Bedrooms" as shown in Figure 2. Figure 3 shows a **list of user records** shown to match the user's search of the predesignated database fields, thus **a user record** of the search criteria **list**. Exh.1004-¶194.
13.	... claim 12 wherein said server memory includes program codes for directing said server processor **to transmit said compressed file to each of said user addresses identified in said user address fields of said user records on said distribution list**.

	{34,54}
	See Claim 8 for reference reciting program codes which direct the server processor to produce a compressed file. Exh.1004-¶195. A POSITA would understand that if there are program codes to direct a server processor to produce a compressed file, then upon execution of those program codes, the server process would serve a compressed file. A POSITA would further understand that there would be program codes to direct the server to whom to serve the compressed files. Exh.1004-¶197.

The above claims are obvious over Peterson, Jensen, Papaj in view of Surfas and Chapman.

The ORACLE database manager disclosed by Papaj discloses a system with server memory that includes programs that create compressed files. Exh.1004-¶¶177-182. A POSITA would also understand these compressed files could be included in or with the message. Exh.1004-¶187.

Chapman also discloses program codes that provide compression with standard Web servers, such as Apache. Exh.1004-¶181. A POSITA would also understand these compressed files could be included in or with the message. Exh.1004-¶187.

Surfas discloses that image files, such as JPEG files, are compressed files. Exh.1004-¶183.

Jensen discloses using the Web interface to show an image file to the user. Exh.1004-¶¶185.

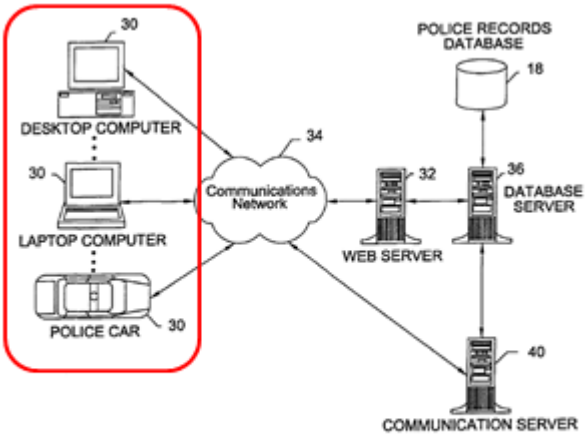
Peterson discloses tracking a user via a Web application and transmitting a message to the user. Exh.1004-¶¶189-193. This requires that the Web application have a user address to which to send the message. *Id*. Peterson also discloses the user executing a query that will scan the information records for a set value (*e.g.*, price or number of bedrooms). Exh.1004-¶¶194-195.

Because Surfas and Chapman each further discuss the broad functionalities of Web servers, WebHub, and Delphi, and are explicitly referenced and cited by Peterson, a POSITA would view these references as already combined because together they disclose the multiple facets of a single complete system. *Id*.

D. Ground 4

Brown, Gladney, and Linstead render claims 1{22,42,62},2{23,43}, 21,63{77,88,99} and 100 obvious under §103 (*see also* Exh.1014-00172--00194):

1.	A **remotely updatable database system** comprising: **{22,42,62,100}**
	"**an SQL database**" and "by other suitable relational **database** applications such as **Oracle**." Exh.1016-0014, 10:4-11. "**database** module is a SQL **database**." Exh.1016-0014 at 9:49–50 (referring to Microsoft SQL Server) In the mid-1990s, both Oracle Database and Microsoft SQL were themselves client/server systems capable of accepting updates from remote clients. Exh.1014-¶184. "when said police record **database** is **updated**." Brown, Claim 14; Exh.1014-¶186.

	"information or police **reports** in a police **database**, **including newly entered reports**." Exh.1016-0010, 2:52–54. The "source database" of Gladney is managed by SQL/DS. Exh.1017-0011. The SQL/DS RDBMS, subsequently marketed as "DB2," ran on IBM mainframe computers and accepted remote updates. Exh.1014-¶187.
1(a)	a **user interface** for **communicating with at least one subscriber system** to **receive user input** from a user at said at least one subscriber system; **{22,42,62,100}**
	Brown Figure 2 depicts a web server (32) as the front end to a database server (36) and clients (30) connected to the web server via a network (34). Exh.1014-¶188; Exh.1016-003. "FIG. 2 … for **implementing the system** 10 is illustrated. … the **user interface 10 module** 12 can be implemented via a **user interface device** 30 which **receives query information from a police officer** …" Exh.1016-0012, 6:7–12.  FIG. 2. Exh.1016-003. The **User Interface Module** is explicit in Brown Figure 1. Exh.1014-¶189.

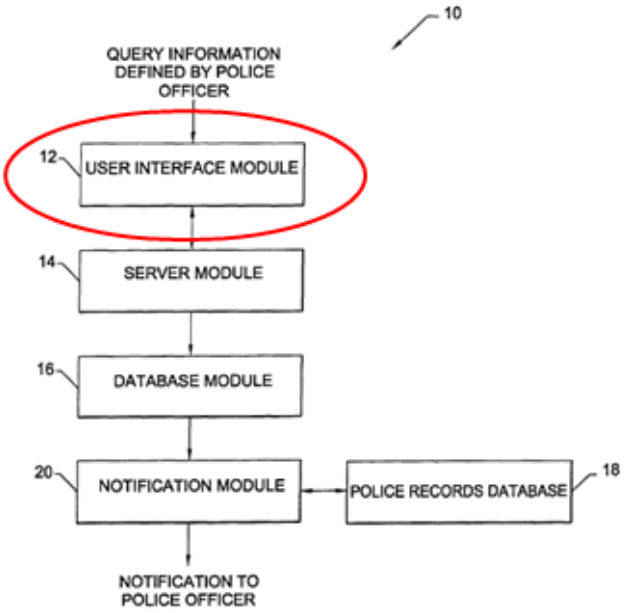


FIG. 1.

Exh.1016-002.

"**user interface module** that receives **user inputted** query information" (claim 1).

"**user interface module** comprises a web page form configured to receive said **user inputted query information through an Internet browser program**" (claim 2).

"**receiving user input** query information from a user" (claim 18). Exh.1014-¶190.

Brown Figure 3 client computer running a Web browser (60) rendering a "web page form" (64), corresponds to "**user interface.**" Exh.1014-¶191.

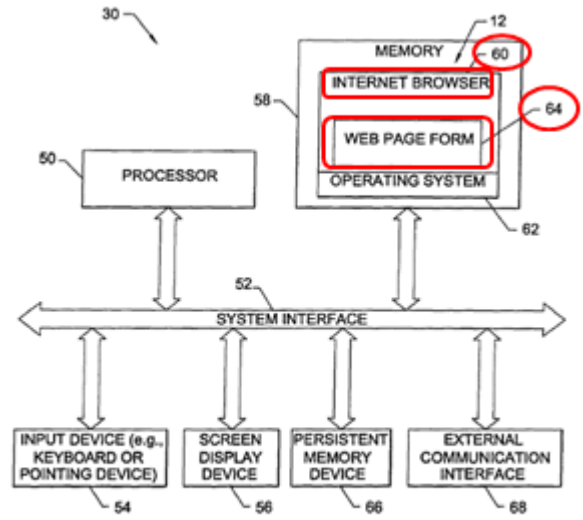


FIG. 3.

Exh.1016-004.

Brown discloses a user interface receiving user input from at least one subscriber system. Exh.1014-¶194. *See, e.g.*, Exh.1016-0010, 2:67–3:1 ("a **user interface module** that receives **user inputted** query information"); Exh.1016-0012, 5:44–46 ("**system 10 comprises a user interface module** 12 that **receives query information from a police officer**"); Exh.1016-0012, 6:47–50 ("a **user interface** device… **communicates** with other elements within the **user interface** device..."); Exh.1016-0013, 7:18–21 ("**police officer can log into the system 10 through any user interface device…**"); Exh.1016-0013, 8:1–50 ("officer **inputs data** for different data fields, …submits the form, which is then r**eceived by the web server and processed**.")

Gladney discloses conventional client/server RDBMS technology in which users at "editor" programs can enter data into forms that are submitted to the server. **User input** is typically received at a client workstation's **user interface** prior to being formatted into a SQL statement and sent to the RDBMS. Exh.1014-¶195. *See, e.g.*, Exh.1017-0020 (editors are described under a heading of "Application Program Responsibilities;" applications, as distinguished from system programs, are typically built for end-users and have a user interface.).

Linstead similarly discloses user terminals communicating with a master DBMS. Exh.1014-¶195. *See* Exh.1018-002 (FIG. 1); *see also* Exh.1018-006, 5:43-47 (regarding the graphical user interface of the Apple

	Macintosh).
1(b)	a **database of information records and user records**, said information records having respective **pluralities of fields**
	"predefined **profiles are established by police officers**, and may include one or more request terms (*i.e.*, request data), a frequency term and **notification information for the officer who defined the profile**." Exh.1016-001, Abstract. "[A] server module which receives the **user inputted query information from the user interface module** and transforms the user inputted query information into a **database procedure**. A **database module executes the procedure against a police record database**, and **generates a notification signal** if a match to the search query occurs." Exh.1016-0011, 3:2–7; 3:43–48. Brown discloses that user records of search criteria are compared to information records at defined intervals. Exh.1014-¶197. See Exh.1016-0015, 11:62-12:7. **User records** include the search query records associated with a police officer user through a username/password log in. Exh.1014-¶198; Exh.1016-0013, 7:6–25. A police officer provides notification information associated with the crime profile such that officer is notified when a match with the crime profile occurs. Exh.1014-¶198; Exh.1016-0010, 2:55–63. In Anderson's data model, 17:20-19:50, every table has at least two columns, disclosing a record with a plurality of fields. Exh.1014-¶199; Exh.1019-0024, 17:20-19:50. Gladney discloses "Workstation Differentiation" in which attributes such as "in California" and "is a pharmacy" are stored. (Exh.1017-0016, Section 5.2). In a late 1990s implementation of Gladney, in which a network-connection personal computer was owned and used by one person, this would function as a "user record." Gladney discloses tables with a plurality of fields. Exh.1014-¶199.

	SELECT some fields FROM SOMETABLE WHERE field1 = :value1 AND field2 = :value2 AND . . . Exh.1017-0020.
1(c)	and wherein **each of said information records is linked to at least one user record**; **{22,42,62,100}**
	A POSITA would have added a user profile data model with user records linked to it. Exh.1014-¶199. Brown also discloses search query records being **associated with a police officer user through a username/password log in.** Exh.1014-¶200; Exh.1016-0013, 7:6–25. "A police officer **provides notification information associated with the crime profile** such that a notification is sent to the officer when a match on the crime profile occurs." Exh.1016-0010, 2:55–63. Gladney discusses using "a single attribute tagging each source record" to correspond to potential workstation differentiation tags, e.g., "in California" or "is a pharmacy." Exh.1017-0016. The only disclosed record storage mechanism in the server of Gladney is SQL/DS, a conventional RDBMS which links records if they share a common field value, and thus the matching of a common attribute would constitute the "linking" claim element. Exh.1014-¶203.
1(d)	a **database manager** in communication with said user interface, **for controlling said database** such that each information record is associated with at least one user, **{22, 42, 62, 100}**
	"[A] server module which receives the **user inputted query information from the user interface module** and transforms the user inputted query information into a **database procedure**. A **database module executes the procedure against a police record database**, and **generates a notification signal** if a match to the search query occurs." Exh.1016-0011, 3:2–7; 3:43–48.

	All records in Brown associated with users logging in and specifying queries are "associated with at least one user." A record regarding a stolen vehicle with a certain license plate would conventionally be "associated with at least one user" (*i.e.*, the user who created it) as soon as entered into the database (see the CREATE_DATE and CREATE_USER fields associated with many tables in Anderson (Exh.1019-0026, 17:20-19:50) for this standard practice). The record would also be associated with a searching user as soon as a police officer entered a query for the same license plate number. Exh.1014-¶204. In Linstead's business information system for recording employee approval of purchase orders (Exh.1018-004, 2:49-58), it would be necessary to associate records with users to accomplish the functions described. Exh.1014-¶205.
1(e)	and **for amending said information records in response to user input** received at said user interface from said at least one subscriber system; and **{22, 42, 62, 100}**
	Brown discloses the use of conventional RDBMS products, such as Oracle and Microsoft SQL Server, that allow users to update or amend police records from remote systems such as desktop personal computers. Exh.1014-¶206. Unlike Colgan (US 5,510,978) and the need for users to check for updates, Brown's improved system allows for remote updates to the database "in response to user input" *and* automatic notifications of those updates to the users. Exh.1014-¶207; Exh.1016-0010, 2:11–21. Gladney's database information records may also be amended by any of the multiple "editor" programs described. Every time a terminal makes a new request to the server a record is added to tables associating that request with a hashcode and timestamp, for example. Exh.1014-¶208; Exh.1017-0011, -0012. Gladney discloses not only the addition of new records, but also changes to several source records, *e.g.*, on Page 80. Exh.1014-¶208.

	Figure 4 illustrates this. A source record $\mathbf{S}_j$ may be used to construct several replica records $\mathbf{D}_{j1}, \mathbf{D}_{j2}, \ldots$. An editor may require that changes to several source records $\mathbf{S}_j, \mathbf{S}_k, \ldots$ be reflected atomically in each cache. For such an update, $\langle \mathbf{S}_j, \mathbf{S}_k, \ldots \rangle$ is the consistency set in the source domain, and $\langle \mathbf{D}_{j1}, \mathbf{D}_{j2}, \ldots, \mathbf{D}_{k1}, \mathbf{D}_{k2}, \ldots \rangle$ is the consistency set in the replica domain. We acquire the needed mapping by recording which portion of the source data is needed for each response built. When we change source records, this dependency relation identifies which prior responses may be obsolete. *The dependency relationship relates query predicates to requests and is independent of source data versions.* Query predicates are used rather than functions of query results to avoid overlooking data insertions. Exh.1017-006. Note the last sentence suggests records can be deleted or inserted as well. Exh.1014-¶209. Exh.1017-0012 below describes deletions: Updating the source data is effected by an editor not depicted in Figure 5. This editor passes the hash code of each deleted or inserted source record to the obsolescence manager. (To satisfy the database history constraint, deletions are managed by inserting new records with a distinguished value in some nonkey field.) For updates, the editor passes the hash codes of both the old and new record versions. It also supplies each changed record with a timestamp, using a common value for every record altered within a transaction. When all the updates needed for a consistent source change have been made, the editor again calls the obsolescence manager to inform it that a request consistency set may be computed.
1(f)	a **message server** in communication with said database manager **for serving a message including at least one record received from said database manager to said at least one user associated with said information record,** **{22,42,62,100}**
	Brown "**sends the appropriate message to the police officer** indicating that there was a match to the submitted query." Exh.1016-0015, 11:3-4. Brown uses the popular Microsoft Exchange Server, which communicates with "**database module**." Together these correspond to the '908 Patent's "message server" and "database manager." *See* Exh.1016-0012, 6:20-26 ("**If a match occurs** during execution of the procedure, then the database module 16 **generates a notification signal** which is passed on to the

notification module 20 which can be implemented via a communication server 40. The communication server 40 generates the appropriate notification message which is **sent to the police officer, or a group of police officers**.") Exh.1014-¶211.

Exh.1016-0011, 3:49–50: "A notification module may receive the notification signal, and may generate a notification message to a user in response to the notification signal.");, 3:65–67 ("If a match to the search query occurs in the step of executing the database procedure, then generating a notification **message to the user** or a group of users.") Exh.1014-¶212.

Brown Figure 8 illustrates **database module sending**, and the communication **server receiving**, notification **messages** (190) triggering the communication. The notification would go to the user with whom the triggering record is associated. Exh.1014-¶212.

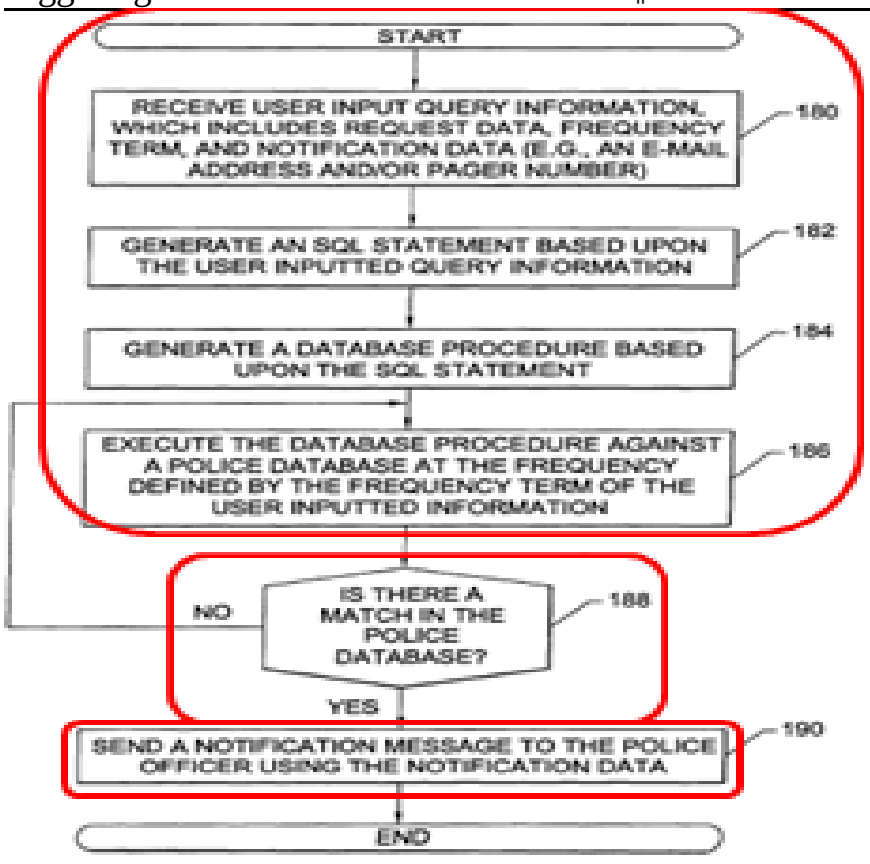
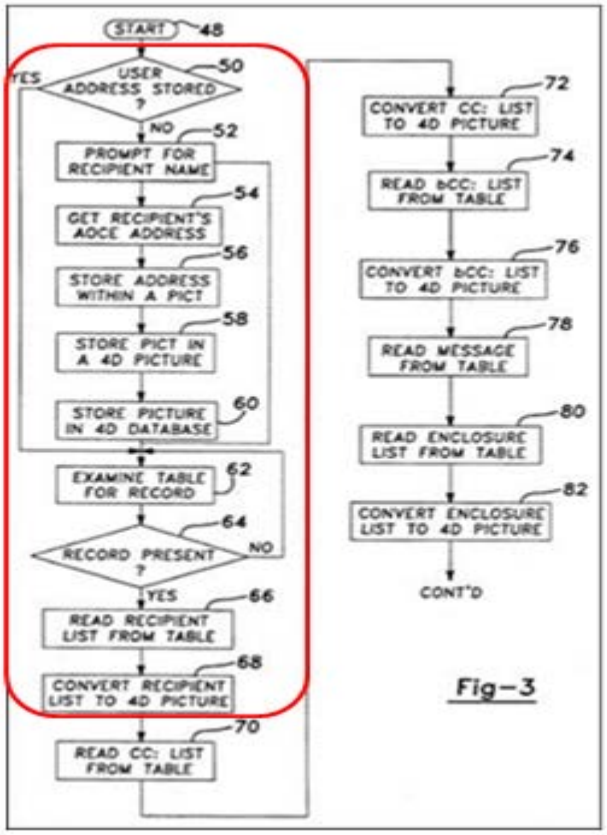


FIG. 8.

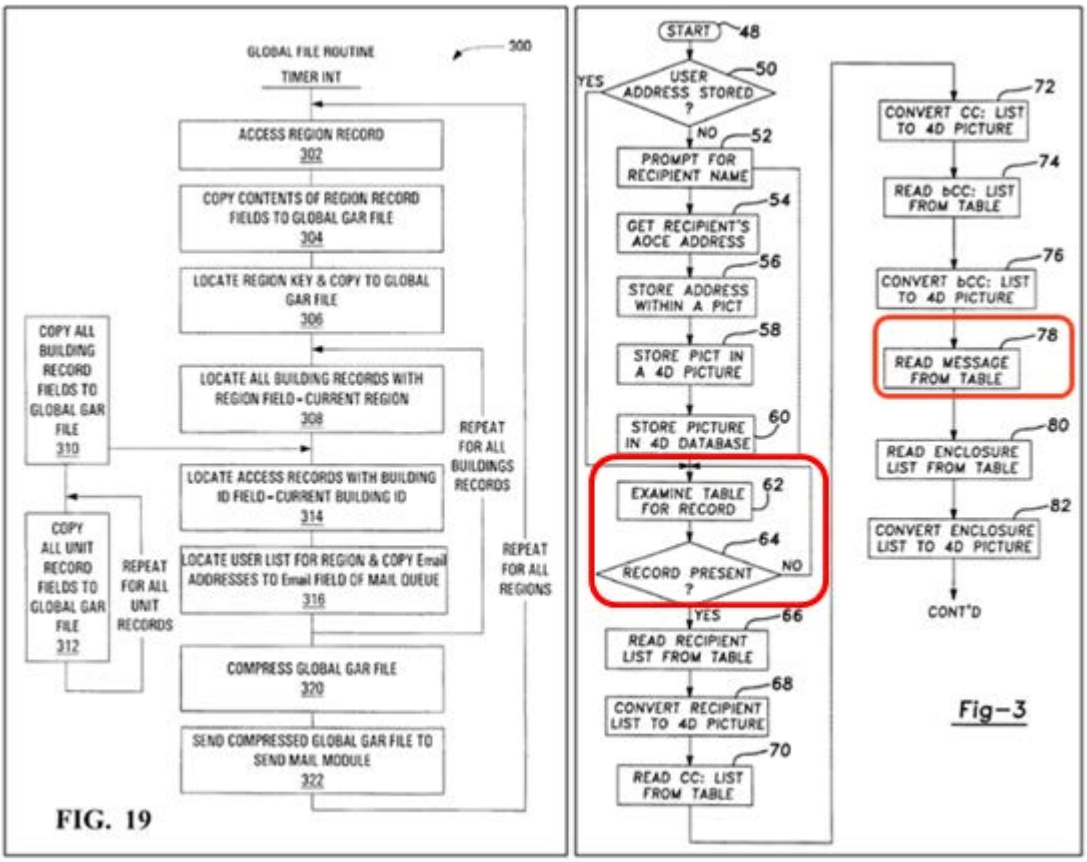
Exh.1016-009.

	"It is then determined whether there is a match in the police records database, as indicated by block 188. If there is no match, the procedure will continue to execute at the frequency defined until a match occurs or the query expires. If there is a match in the police records database, then **a notification message is sent** to the police officer, as indicated by block 190." Exh.1016-005, 12:1 –7. Exh.1016-0014, 10:61-67: "The **notification message may include information from the police record that matched the search query so that the police officer is provided with substantive information in the notification message**. Further, the notification message may include a search ID or record number so the police office can log into the system 10 for more information regarding the search results." Exh.1016-0012, 6:21-24: "The communication server 40 generates the appropriate notification **message** which is sent to the police officer, or a group of police officers." Linstead's Figure 3 clearly discloses **associating the users with the information records**. See especially steps 50 – 68 below, with particular attention to step 66 in which user email addresses are read from a database record. Exh.1014-¶215.

	 Exh.1018-003. Exh.1018-005, 3:19–24: "The daemon process **detects this record**, reads the record, and **prepares an electronic mail messag**e. The message is passed to the electronic mail routines that are a part of the Macintosh operating system." Exh.1018-008, 9:24–29: "the method allows use of a common data transport mechanism, electronic mail, that in turn allows all electronic mail users, whether local or remote, **to receive information concerning data events and results taking place within a database environment**."
1(h)	said message server including **a processor and memory for storing program codes** readable by said processor **to direct said processor to communicate with said database manager** to obtain for inclusion in said message a plurality of information records **having at least one common field entry.** **{22,42,62,100}**

The stored procedures of Brown connect to notification module (20) on "communication server" (40), which may be running a conventional email server programs like Microsoft Exchange. Exh.1016-0015, 11:10-46. Exh.1016-0012, 5:15–20 discloses **computer program instructions** "**stored** in a computer-readable **memory** that can direct a computer to other programmable **data processing apparatus** to function in a particular manner." Exh.1014-¶216.

Linstead discloses a continuously running "daemon process" that functions as the claimed "message" server. This process is disclosed as "running in a loop" (steps 62-64 below right) to detect new records in a table of alerts. Compare the flow diagrams of Figure 19 of the '908 Patent (Exh.1001-003, below left) and Fig-3 of Linstead (Exh.1018-003, below right). Exh.1014-¶218.



Linstead discloses a message (78 'read message') to be included in an email. Linstead discloses the use of 4D, an RDBMS for storing tables with

	a plurality of information records having at least one common field entry. Exh.1014-¶218. A POSITA would understand from Fig-1 of Linstead, showing a DB at every user workstation, and the description at Exh. 1018-007, 8:23-24 of "computer files having a predetermined format" being sent via email that Linstead was sending information records in emails. Exh.1014-¶219-220 Exh.1018-008, 9:11–13; claim 1: "a method for automatically providing an indication of the occurrence of a predetermined event **within a database system** to one or more users of the database system." Exh.1018-007, 7:18-26 describes communicating a purchase order to a supervisor, with the transmission being from one computer running the 4th Dimension database management system to another. A purchase order for multiple items would conventionally be represented in an RDBMS such as 4D with multiple rows, linked with a common field entry such as "purchase_order_ID". Exh.1014-¶219-220. Gladney explicitly discloses that records in the source database as "interrelated," Exh.1017-005. Gladney relies on SQL/DS, a conventional RDBMS, and therefore "interrelated" record would share a common field entry. When there are changes to interrelated records, they are sent as a "consistency set" in a single message Exh.1017-0013,-0014. Exh.1014-¶224.
2.	... claim 1 wherein said message server includes a processor and memory for storing program codes readable by said processor to direct said processor to communicate with said database manager to obtain for inclusion in said message **at least one** information record having at least one common field entry. **{23,43}**
	Every limitation in Claim 2 is already present in claim 1. See Claim 1 above. Software that can send multiple information records can also send "at least one" record. Exh.1014-¶225. The claimed additional element of a memory storing program codes is shared by all modern computers. Note that the claimed feature of a memory storing program codes readable by a processor is inherent in any software-based system built since the 1950s. The "stored program" architecture, in

	which the program to be run by a computer is stored in the computer's memory (as opposed to being implicit in patch cables), is shared by all modern digital computer systems. Exh.1014-¶226. Linstead's "stored program codes" are the program codes, executable by the Macintosh computer of Linstead, that can look for a record in the 4th Dimension database and prepare the email messages, with attachments, as described in Exh.1018-007, 7:29 through Exh.1018-008, 9:9. Exh.1014-¶228. Brown sends information from database records at least in a user-readable format. Exh.1014-¶227. *See* Exh.1016-0010, 2:64-67, Exh.1016-0011, 3:1-5 ("a law enforcement data analysis system for automatically monitoring police records for a crime profile may comprise a user interface module that receives user inputted query information defining a search query for searching for the crime profile, and a server module which receives the user inputted query information from the user interface module and transforms the user inputted query information **into a database procedure**."). A POSITA would understand that information records were being sent in "computer files having a predetermined format" (Exh.1018-007, 8:23-24) and that these would typically include related records, thus meeting the "common field entry" limitation. Exh.1014-¶228.
21.	... claim 1 wherein said user interface includes a **web interface**.
	The only interface described in Brown is a web interface, *e.g.*, FIG. 4 and FIG 2 (note the "Web Server" front-end to the "Database Server"). Exh.1014-¶229. "user **interface** module **comprises** an **web page** form." Exh.1016-0016, 13:5-8. "…. **user interface** device 30 is a client side device provided to execute the **user interface** module 12 which, in a preferred embodiment, **comprises a web page form** (hereafter referred to as a query information form) displayed through an **Internet browser** program." Exh.1016-0012, 6:35-39.

63.	An apparatus for receiving database record information from a database containing a plurality of database records, the apparatus comprising: a) a message receiver **for receiving a message from said database**, said message including at least one file from said database; b) transfer memory for **storing said at least one file**; c) an output device for **presenting information to a user**; and d) a processor circuit in communication with said message receiver, said transfer memory and said output device, and processor memory programmed with processor-readable codes for directing said processor to direct said output device to present to the user a list of files stored in said local memory, the files included in said list **having at least one predefined characteristic**; and to direct said output device to present to the user at least some of the contents of at least one file having said at least one predefined characteristic. **{77,88,99,100}**
	See Claim 1 above. Claim 1 has no utility unless there is a corresponding client to receive the non-standard messages that this server of Claim 1 sends. Brown describes a "thin client" system (*e.g.*, online web browser with little to no custom software). *See* Exh.1016-004 (FIG. 3). The passage in the '908 Patent that corresponds to claim 63 discloses a simple HTML page offering choices to an end-user. Exh.1001-0040, 14:29-45. Brown's FIG. 3 would be able to meet these claim limitations via receiving a page from an appropriate server, *e.g.*, a user checking the order history from amazon.com circa 1997. This would result in the receipt of a dynamically generated page from Amazon merging database contents with an HTML template, with option to click on a given order and see additional detail. The element of the "transfer memory" corresponds to the memory illustrated in FIG. 3 and to any memory conventionally used by a browser program in its rendering operations. Exh.1014-¶235. "Examples of commercially available middleware application suitable for implementing as server module include **WebObjects environment** [*e.g.*, U.S. Patent No. 6,249,291] by Apple Computer Inc." Exh.1016-0014, 9:16-

18.

Figures 3 and 5 of Gladney show a "fat client" (*e.g.*, offline PC with custom software that connects to a RDBMS) with user interface presentation capability (the "presentation managers"). Each workstation has a cache of database records that have been received in messages from the server. Exh.1014-¶236.

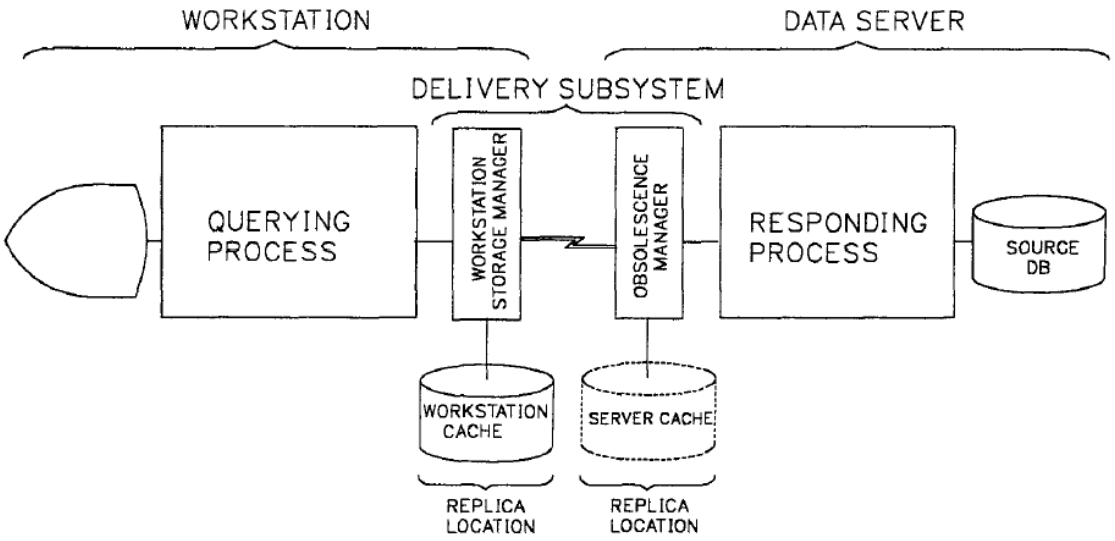


Fig. 3. Model of a single application session.

Exh.1017-005.

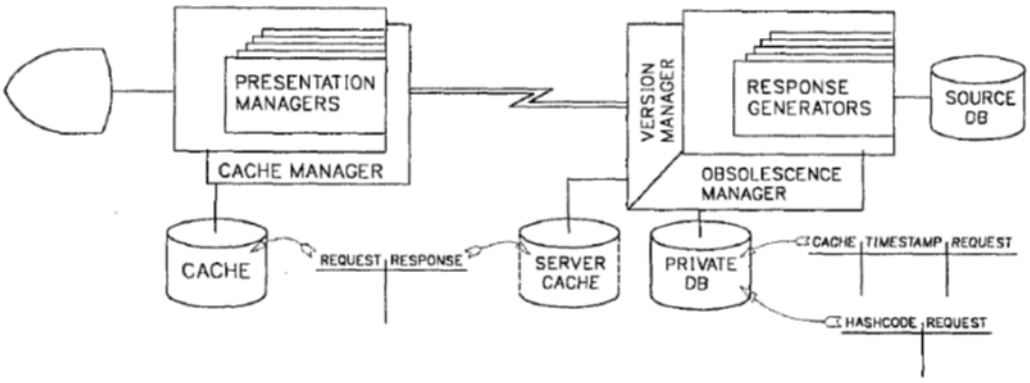
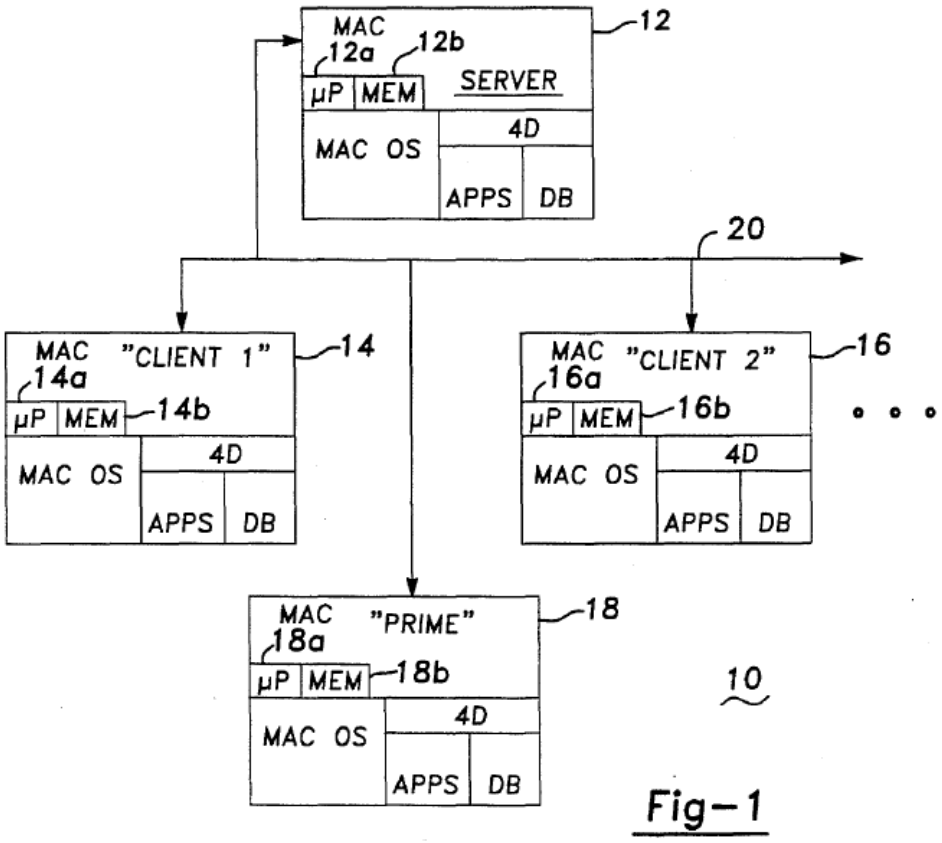


Fig. 5. Prototype configuration.

Exh.1017-0012.

Figure 1 of Linstead illustrates clients, each of which runs a copy of the 4D

	DBMS software, capable of storing records locally in memory, and presenting information from those records to a user (including structured information like email). Exh.1014-¶239.  Fig–1 Exh.1018-002.
100.	Claim 100 is an obvious and verbatim combination of Claims 1 (a server) and 63 (a client).
	See Claims 1 and 63 above for references reciting the majority of the claim limitations. Moreover, a central server is typical in a database application, and in any event not novel but completely obvious as of the priority date. Exh.1014-¶247.

A POSITA faced with the problem of distributing database records and record updates to poorly connected remote users would have naturally combined Brown with the distributed database management techniques provided in Gladney and Linstead.

Brown teaches a web-based RDBMS that accepts updates and queries from remote users through a user interface. Exh.1014-¶¶184-186. Brown's database has user records (police officers and their queries) and information records (police records regarding crimes). *Id*. Each record has multiple fields identifying different pieces of data (*e.g.*, a frequency term, notification information, crime profile). Exh.1014-¶¶196-99. Brown discloses running searches across the database to match information records with user search criteria and sending notifications of matches through its message server. Exh.1014-¶¶201-04. Brown's system improves on the prior art and eliminates the need to affirmatively check for updates in a database. Brown's system allows users to update the database remotely and receive automatic updates of those database changes. Exh.1014-¶206.

The police officers in Brown were geographically dispersed, as were the client workstations in Gladney, and each client in Brown may have sought different information, also as in Gladney. Exh.1014-¶¶176. Gladney teaches the ability to direct records to specific users based on ATTRIBUTES (tags) on records and workstations as well as "consistency set." Exh.1014- ¶¶177; 226.

A POSITA would have been motivated to modify Brown to produce a system in which users could receive alerts without being continuously connected to the Internet and then explore the information within those alerts without an Internet connection. Exh.1014-¶177. Brown recognized that network connectivity, even for mobile users, was improving dramatically around 1997. Exh.1014-¶174. A POSITA faced with the task of implementing notifications from a database with larger records (*e.g.*, photographs) or faced with using an intermittently connected network such as dialup, would have turned to distributed database management system techniques, such as those disclosed in Linstead. *Id.* Linstead discloses a DDBMS that responds to database updates by packing up information retrieved from the database and sending it to clients via emails with actual "information records." Exh.1014-¶¶176,178. While Linstead and Gladney are client/server systems using pre-web technology, a POSITA would have ample reason to adapt these existing systems with a web interface as disclosed in Brown. Exh.1014-¶¶183,231-232. There were tens of millions of potential users worldwide with installed web browsers, and thus, starting in the mid-1990s it was conventional throughout the world of corporate IT to convert client/server systems to systems with web front-ends. Exh.1014-¶237.

It would be obvious to a POSITA to enhance the system disclosed in Brown with the distributed database solution disclosed in Gladney, a solution from IBM

(the world's leading vendor of business information technology in the mid-1990s), which is directed at a similar challenge of supplying workstations with the most relevant updates from a central database to remote users. Exh.1014-¶174.

E. Ground 5

Brown, Gladney, Linstead, and Anderson render claims 3-17{24-38,44-59} obvious under §103 (*see also* Exh.1014-00195--00201):

3.	... claim 1 wherein said database manager stores the contents of at least some of said information records in a **compressed format**. **{24,44}**
	Anderson discloses associating two-character DIET_CODE with a CARD_NUM in the HH_DIET table (see above), rather than repeatedly storing the 30-character DIET_DESC field. *See* the CARD_MEMBER table (Exh.1019-0026, 17:28-47) in which the EMPL_STATE column is limited to two characters (Exh.1019-0027, 20:25), storing "MD" rather than "Maryland." Exh.1014-¶252. The '908 patent also discloses using the standard technique of bit-fields, *e.g.*, in Figure 8A (Exh.1001-0011) with accompanying description in Exh.1001-0039, 11:33-63. The use of bit-fields (also known as "bit vectors" or "bit sets") is explicitly disclosed in the C++Library as well as in descriptions of standard protocols such as TCP/IP. Exh.1014-¶253. Linstead discloses storing in the database "**digital representations of audio, video, pictures.**" Exh.1001-0021, 8:25. As these files are enormous and impractical to handle, especially with the storage systems of 1994, the standard 1994 formats for storing audio, video, and pictures included data compression. Exh.1014-¶254.
4.	... claim 3 wherein said contents of at least one field of said information record includes at least **one information code**. **{25,45}**
	Anderson discloses the use of a two-character DIET_CODE column that

	meets this limitation, the use of a two-character EMPL_STATE code, and the use of a two-character INCOME_CODE representing an income range. Exh.1014-¶¶255-256.
5.	... claim 4 wherein said at least one information code represents at least one of a **plurality of predefined contents of said field**. **{26,46}**
	The EMPL_STATE field in Anderson is a predefined (by the U.S. Postal Service) list of valid state codes. Additionally, the DIET_CODE field of the HH_DIET table represents a reference to one of the predefined records of the DIET_CODES table. Note that this practice is conventional in the RDBMS world and standard systems, such as Oracle, provide support for ensuring referential integrity, *e.g.*, that a DIET_CODE cannot be inserted into the HH_DIET table unless there is a corresponding predefined record in the DIET_CODES table. Exh.1014-¶257. Linstead also makes use of this conventional relational database practice. Exh.1014-¶258. "Thus, the indication of the **occurrence of the predetermined event** within the database system includes at least **a portion of the record or data structure written to the predetermined storage location**. Alternatively, however, the record could only indirectly identify the message and its recipients, for example, by consisting of a **pointer to a storage location containing a predetermined message and group of recipients**." Exh.1018-007, 7:57-64.
6.	... claim 4 further including **memory for storing a key** identifying said predefined contents of each of said fields represented by said at least one information code. **{27, 47}**
	The DIET_CODE of Anderson, discussed above, is both an "information code" and a key (to the DIET_CODES table). Exh.1014-¶259.
7.	... claim 6 wherein said server memory includes **program codes for directing said server processor to produce a composite file** including said key and at least one of said information records. **{28,48}**

	The message of Gladney contains multiple rows from an RDBMS (a "consistency set") and is a "composite file" of information records. Exh.1014-¶262. The PICT-format file that is produced by the server of Linstead prior to sending email to users would typically be an example of the claimed "composite file". Exh.1018-007, 8:16-26. It is inherent that the messages and files sent by the system of Linstead include database keys because the contents of the email messages are imported into 4th Dimension databases on the client computers. Exh.1014-¶260. Linstead discloses the use of a standard stored-program computer, *e.g.*, Apple Macintosh, in which any instructions to be executed by the server's processor must be stored in the server's memory. Exh.1014-¶261. "The processor operates in response to **programs of instructions stored in the memory**." Exh.1018-004, 1:40-42.
8.	... claim 7 wherein said server memory includes program codes for directing said server processor to produce a compressed file **including said key and at least one of said information records**. **{29,49}**
	See Claim 7 above for references reciting the majority of the claim limitations. Exh.1014-¶263. Linstead gives an example of a situation in which the database stores "inventory information, including a list of stocked parts" and thus a POSITA would infer that a conventional database key such as a part number was employed. Exh.1014-¶263.
9.	... claim 8 wherein said **compressed file includes an image file**. **{30,50}**
	Exh.1018-007, 8:23-26: The ***enclosures*** may be computer files … consisting of digital representations of audio, video, ***pictures*** or text data. … the ***picture*** containing the enclosure data is read and the ***enclosure*** data is extracted from the ***picture***. … electronic mail message is formatted

	using … the enclosure data. (Emphasis added.) Gladney discloses the use of images of items in a "sales catalogue" (page 85) and of plants (page 84). Exh.1017-0011,-0010. Gladney suggested trying to avoid, circa 1989, transmitting the largest files over the 1200-baud modem in use at the time. However, a POSITA would have been motivated to make appropriate use of improved networking hardware in 1998 by sending images over the network rather than in postal mail. Exh.1014-¶265.
10.	... claim 9 **wherein said message includes said compressed file**. **{31,51}**
	See Claim 8 for reference reciting program codes which direct the server processor to produce a compressed file. Exh.1014-¶267. Exh.1017-007, 8:23-40 discloses that a single message is sent including enclosures that contain compressed files, namely images.
11.	... claim 10 wherein said user records include a **user address field** for storing a user address to which a message produced by said message server is to be transmitted. **{32,52}**
	Brown Figure 4 shows that a user can enter an e-mail address into a form; Exh.1016-0014, 9:22-46 explains how the form-submitted data is recorded in the database; Exh.1016-0015, 11:21-25 discloses using the Exchange email server, from which a POSITA would infer that an email address had been previously stored among the user records. Linstead explicitly discloses using electronic mail (Exh.1018-006, 5:10-12), storing user email addresses (Exh.1018-006, 6:57-62), and working with previously stored email addresses (Exh.1018-007, 7:48-53). Exh.1014-¶268.
12.	... claim 11 wherein said memory includes program codes for directing said processor to direct said database manager to **scan said user records to obtain a distribution list of user records** having the contents of at least one field associated with the contents of a predesignated field in at least one of said information records. **{33,53}**

	Brown discloses that user records of search criteria are compared to information records at defined intervals. User records include defined stored procedures. Exh.1014-¶197. See Exh.1016-0015, 11:62-12:7. When a match between the information sought by the police officer and the contents of a database record is found, an alert message is sent out. Exh.1014-¶269. Linstead takes the alternative approach of processing one information record at a time and, given that record, finding the list of users that need to be notified. Exh.1014-¶270; Exh.1018-007, 7:40-53.
13.	... claim 12 wherein said server memory includes program codes for directing said server processor **to transmit said compressed file to each of said user addresses identified in said user address fields of said user records on said distribution list**. **{34,54}**
	See Claim 8 above for reference reciting program codes which direct the server processor to produce a **compressed file**. See claim 11 above for **address field** for storing the user's address to where the message is to be transmitted. See claim 12 above for **distribution lists**. Exh.1014-¶272.
14.	... claim 4 wherein said at least one information code represents a **quantitative measure of a predefined physical quantity**. **{35,55}**
	Linstead discloses that the database may include "an associated list of **on-hand quantities of each stocked part**." Exh.1018-006, 6:3-6. Anderson shows the representation of a physical quantity, *e.g.*, the quantity of an item that was purchased in the TOT_PROMO_QTY column of the HH_PROMO_PURCH table. Exh.1014-¶274; Exh.1019-0026, 18:42-47. Linstead also discloses storing thresholds for these quantities: "When inventory falls below a **predetermined level** which is defined in the application, the application prepares a purchase order in response to a low inventory condition." Exh.1014-¶273; Exh.1018-006, 6:20-23. Anderson discloses database fields for recording quantities of goods

	purchased, *e.g.*, TOT_CLUS_QTY (Exh.1019-0026, 17:64) and TOT_PROMO_QTY (Exh.1019-0026, 18:43). Exh.1014-¶274.
15.	... claim 14 further including **memory for storing a key identifying said predefined physical quantity**. **{36,56}**
	See Claim 6 for references reciting memory for storing a key. Exh.1014-¶275. In systems where users are expected either to enter data as a range or query by predefined ranges, a POSITA would typically create a helper table with ranges and associated codes or keys for those ranges. Exh.1014-¶275. Examples of such table are provided in U.S. Patent 5,974,396 ("Anderson '396", Exh.1019-0025), a system for analyzing consumer purchasing behavior. For example, INCOME_CODES is described in column 15 as "A codes [sic] list of consumer household income ranges" (See also column 18 for the actual columns in this table.) Exh.1014-¶275.
16.	... claim 15 wherein said key includes a **character string** associated with said field. **{37,57}**
	Using the INCOME_CODES is described in column 15 as an example again, it is clear that Anderson discloses the use of character strings because the INCOME_CODE is a 2-character field. Exh.1014-¶276; Exh.1019-0027, 20:37.
17.	... claim 4 wherein said at least one information code identifies whether or not any of a plurality of pieces of information are to be **associated with** said record. **{38,58}**
	The above-referenced DIET_CODE field of the HH_DIET table in Anderson discloses "one of a plurality of predefined contents of said field" through the multiple columns of DIET_CODES. Thus a two-character value for DIET_CODE in the HH_DIET table not only associates the 30-character DIET_DESC with a loyalty card, but also associates the additional values in the CREATE_USER and CREATE_DATE fields of

	the DIET_CODES table. Exh.1014-¶277. *See* above and Exh.1019-0026, 17:55-57 and 18:41-42 for the structure of the DIET_CODES and HH_DIET tables.

The addition of Anderson to the above disclosures renders these dependent claims obvious.

Anderson discloses compression (as the term is used in the ’908 patent) and using an additional table, which contains ranges for quantity and codes (or keys) for those ranges. Exh.1014-¶¶253-254. Linstead also discloses data compression in the form of storing digital representations of media files. Exh.1014-¶¶256-257. Linstead and Anderson both disclose abbreviating an information record to an information code. Exh.1014-¶¶257-258. Those same disclosures also demonstrate using the code to represent a plurality of contents. Exh.1014-¶¶259-260.

Both Anderson and Linstead disclose predefined contents of fields. *Id*. In Anderson, various fields in the table represent a plurality of predefined contents of that field. Exh.1014-¶259. Anderson also discloses using a key to identify the fields in the information record referenced by the information code. Exh.1014-¶261.

Linstead further discloses a PICT-format file that is a composite file of a key and an information record. Exh.1014-¶262. Linstead also discloses the use of a standard stored-program computer. Exh.1014-¶263. Gladney also discloses a

composite file of information records containing keys. Exh.1014-¶264. Linstead also discloses the ability to create a compressed filewhich can include a compressed composite file. Exh.1014-¶265. Among the compressed files Linstead discusses, sending a compressed image via a message is disclosed. Exh.1014-¶266. Gladney also discloses the use of images in a networked environment, and a POSITA would understand any images sent would be in a compressed format. Exh.1014-¶¶267-268. Additionally, Linstead further discloses a message sent that contains the compressed files, including files that contain images. Exh.1014-¶269.

Linstead and Brown also disclose storing user e-mail addresses to which messages are sent. Exh.1014-¶270. Brown and Linstead both disclose scanning information records for certain information and notifying specified users when that information is found and in light of Anderson, this could be a compressed file sent to the user email addresses. Exh.1014-¶¶271-274. Linstead also discloses that the information code can include a list of quantities, which shows a quantitative measure of physical quantity of a field. Exh.1014-¶275. Anderson also shows an information code that can include a representation of a physical quantity (*e.g.*, the quantity of an item that was purchased). Exh.1014-¶276.

Anderson also shows a table of codes, in which each code acts as a key and has a corresponding predefined quantity to represent the contents of each field. Exh.1014-¶277. In addition, Anderson demonstrates the use of character strings

associated with each field. Exh.1014-¶278. Anderson also discloses one of a plurality of predefined contents of said field, as represented in the aforementioned table. Exh.1014-¶279.

It would have been obvious to a POSITA to further modify the conventional web-based database application of Brown and the known database management techniques of Linstead, to include the information codes, keys, and compression methods from Anderson because it is conventional in the RDBMS world to include helper tables any time a list is represented, (*e.g.*, a list of physical quantity ranges). Exh.1014-¶251. The information codes, keys, and compression methods disclosed in Anderson are basic, standard techniques taught in any freshman data programming class. Exh.1014-¶252. This would have been an obvious programming choice to one of ordinary skill in the art. *Id*.

F. Ground 6

Brown, Gladney, Linstead, Anderson, and C++Library render claims 18-20{39-41,59-61} obvious under §103 (*see also* Exh.1014-00202--00203):

18.	... claim 17 wherein said at least one information code includes **a decimal number representing a binary number having individual bits representing the inclusion or exclusion of a corresponding predefined piece of information**. **{39,59}**
	Extensive support in the popular C++ language has been available for bit sets at least since 1991, as demonstrated by the Bit_Set class in C++Library (Exhibit 1020-00138 through -0040). C++Library, page 8-8 discloses a

	function for testing an individual bit. Exh.1014-¶280. **inline Boolean operator[] (int** *n*) **const**; Returns **TRUE** or **FALSE** to indicate the setting of the zero-relative *n*th bit. If the index is out of range, an **Error** exception is raised. Exh.1020-00139. Multiple bits can be tested at once, *e.g.*, to see how closely a bit patterns aligns to a sought-after pattern (page 8-9). Exh.1014-¶281. **inline void set_intersection** (**const Bit_Set&** *bs*); Determines the logical intersection of *bs* and the bit set object, modifying the source with the result. This function invalidates the current position of the bit set object. Exh.1020-0041.
19.	... claim 18 further including **a key** identifying said plurality of pieces of information. **{40,60}**
	See Claim 6 for references reciting memory for storing a key. Exh.1014-¶287. As noted above and explained in Anderson and C++Library, it is conventional for any database table to include a unique key. Exh.1014-¶287.
20.	... claim 19 wherein said **key includes a plurality of character strings** associated with said field. **{41,61}**
	See Claim 16 for references reciting a key that includes a plurality of character strings. Exh.1014-¶288. As noted above, the main purpose of the standard RDBMS is to permit the recording of arbitrary associations among keys, character strings, and other types of data. Exh.1014-¶288.

Keys and character strings as claimed are all basic foundational tools for relational databases. Exh.1014-¶¶47-57, 281-282. A POSITA would also know that C++Library includes the use of bit field. Exh.1014-¶¶281-282. C++Library also discloses that it was conventional for a database to include a unique key to identify information. Exh.1014-¶289. C++Library and Brown (as relational databases) both disclose using a key that has character strings associated to a field in the relational database. Exh.1014-¶290. Linstead discloses combining multiple fields into one data structure for transport.

A POSITA who found that the same attribute information was being packed into data structures repeatedly would have been motivated to try a bit-field approach. Exh.1014-¶288. A POSITA would combine conventional RDBMS references such as Brown, Linstead, and the C++Library with bit vector processing to facilitate flexible searches. Exh.1014-¶285. An alternative reason why a POSITA might choose to combine conventional RDBMS techniques with bit fields is a shortage of disk space. Exh.1014-¶¶286-287. A POSITA facing a dire shortage of disk space or a need to communicate database records over slow communication links (the problem the '908 Patent sought to solve, Exh.1001-6:65-7:9) would be motivated to combine the RDBMS references with the bit-field support of C++Library. Exh.1014-¶287.

Anyone of ordinary skill in the art creating a web-based database application would have been well aware of all of the recited information code techniques, as evidenced by C++Library. Exh.1014-¶280. C++ would be easily combined with the above references because it is a basic programming language taught to any computer-science freshman. Exh.1014-¶158.

VIII. CONCLUSION

'908 Patent claims 1-63, 77, 88, 99, and 100 are unpatentable. As to all of them, there is a reasonable likelihood that Petitioner will prevail. *Inter Partes* Review of these claims is requested.

Dated : May 19, 2016

Respectfully submitted,

/Yeu-Yan Perng/
Yeu-Yan Perng (Back-Up Counsel)
Tel: (206) 447-6233
Fax: (206) 749-2062
Email: Yan.Perng@foster.com
Registration No.: 74815

/Benjamin J. Hodges/
Benjamin J. Hodges (Back-Up Counsel)
Tel: (206) 447-6282
Fax: (206) 749-2062
Email: Ben.Hodges@foster.com
Registration No.: 69500

/Joel B. Ard/
Joel B. Ard (Back-Up Counsel)
Tel: (206) 447-6252
Fax: (206) 749-2062
Email: Joel.Ard@foster.com
Registration No.: 67041

/P.G. Scott Born/
P.G. Scott Born (Back-Up Counsel)
Tel: (206) 447-6254
Fax: (206) 749-2062
Email: Scott.Born@foster.com
Registration No.: 40523

/Richard T. Black/
Richard T. Black (Lead Counsel)
Tel: (206) 447-6251
Fax: (206) 749-2062
Email: Rich.Black@foster.com
Registration No.: 40514

Attorneys for National Association of Realtors
FOSTER PEPPER PLLC
1111 Third Avenue, Suite 3000
Seattle, WA 98101

CERTIFICATE OF WORD COUNT PURSUANT TO 37 CFR § 42.24

Pursuant to 37 CFR § 42.24, I hereby certify the word count for Petitioner National Association of Realtors' Petition for *Inter Partes* Review of U.S. Patent No. 6,529,908B1 Under 35 U.S.C. §§ 311-319 and 37 C.F.R. § 42.100, et. seq. is 13,644 words in length, which does not exceed 14,000 words as required by the rule, excluding the table of contents, mandatory notices under §42.8, the certificate of service or this certificate of word count, appendix of exhibits or claim listing. I further certify that this Petition for *Inter Partes* Review was prepared using Microsoft Word and this is the word count Microsoft Word generated for this document.

Dated: May 19, 2016

Respectfully submitted,

/Yeu-Yan Perng/
Yeu-Yan Perng (Back-Up Counsel)
Tel: (206) 447-6233
Fax: (206) 749-2062
Email: Yan.Perng@foster.com
Registration No.: 74815

/Benjamin J. Hodges/
Benjamin J. Hodges (Back-Up Counsel)
Tel: (206) 447-6282
Fax: (206) 749-2062
Email: Ben.Hodges@foster.com
Registration No.: 69500

/Joel B. Ard/
Joel B. Ard (Back-Up Counsel)
Tel: (206) 447-6252
Fax: (206) 749-2062
Email: Joel.Ard@foster.com
Registration No.: 67041

/P.G. Scott Born/
P.G. Scott Born (Back-Up Counsel)
Tel: (206) 447-6254
Fax: (206) 749-2062
Email: Scott.Born@foster.com
Registration No.: 40523

/Richard T. Black/
Richard T. Black (Lead Counsel)
Tel: (206) 447-6251
Fax: (206) 749-2062
Email: Rich.Black@foster.com
Registration No.: 40514

FOSTER PEPPER PLLC
1111 Third Avenue, Suite 3000
Seattle, Washington 98101-3296
Attorneys for National Association of Realtors

CERTIFICATE OF SERVICE

The undersigned hereby certifies that a true copy of the foregoing PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 6,529,908B1 UNDER 35 U.S.C. §§ 311-319 AND 37 C.F.R. § 42.100, ET. SEQ. as well as the accompanying Power of Attorney and Exhibits 1001-1048, have been served in its entirety this 19th day of May, 2016 by U.S.P.S. Priority Mail Express No. 302691531 mail upon the following:

Daniel E. Altman
Knobbe Martens
Intellectual Property Law
2040 Main Street, 14th Floor
Irvine, CA 92614

Dated: May 19, 2016

Respectfully submitted,

/Yeu-Yan Perng/
Yeu-Yan Perng (Back-Up Counsel)
Tel: (206) 447-6233
Fax: (206) 749-2062
Email: Yan.Perng@foster.com
Registration No.: 74815

/Benjamin J. Hodges/
Benjamin J. Hodges (Back-Up Counsel)
Tel: (206) 447-6282
Fax: (206) 749-2062
Email: Ben.Hodges@foster.com
Registration No.: 69500

/Joel B. Ard/
Joel B. Ard (Back-Up Counsel)
Tel: (206) 447-6252
Fax: (206) 749-2062
Email: Joel.Ard@foster.com
Registration No.: 67041

/P.G. Scott Born/
P.G. Scott Born (Back-Up Counsel)
Tel: (206) 447-6254
Fax: (206) 749-2062
Email: Scott.Born@foster.com
Registration No.: 40523

/Richard T. Black/
Richard T. Black (Lead Counsel)
Tel: (206) 447-6251
Fax: (206) 749-2062
Email: Rich.Black@foster.com
Registration No.: 40514

FOSTER PEPPER PLLC
1111 Third Avenue, Suite 3000
Seattle, Washington 98101-3296
Attorneys for National Association of Realtors