

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ORACLE CORPORATION,
Petitioner

v.

JOSEPH L. DEBELLIS,
Patent Owner

Case No.: IPR2024-00221
U.S. Patent No. 6,760,720
Original Issue Date: July 6, 2004

Title: SEARCH-ON-THE-FLY/SORT-ON-THE-FLY SEARCH ENGINE FOR
SEARCHING DATABASES

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 6,760,720
PURSUANT TO 35 U.S.C. §§311-319 AND 37 C.F.R. §42**

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. REQUIREMENTS FOR PETITION FOR INTER PARTES REVIEW	3
A. Grounds for Standing (37 C.F.R. §42.104(a))	3
B. Notice of Lead and Backup Counsel, Service Information, and Fee Information (37 C.F.R. §§42.8(b)(3-4), 41.10(a), 42.103)	3
C. Notice of Real-Parties-in-Interest (37 C.F.R. §42.8(b)(1)).....	4
D. Notice of Related Matters (37 C.F.R. §42.8(b)(2)).....	4
III. THE BOARD SHOULD NOT DENY INSTITUTION UNDER §314.....	6
A. Existence Or Likelihood Of A Stay	7
B. Proximity Of Trial Date To Final Written Decision	7
C. Investment In The Parallel Proceeding	7
D. Overlapping Issues	8
E. Identity Of Parties	8
F. Other Circumstances Including The Merits	9
IV. DENIAL UNDER 35 U.S.C. §325(D) WOULD BE INAPPROPRIATE	9
V. THE BOARD SHOULD NOT DENY INSTITUTION UNDER 35 U.S.C. §315(B)	10
VI. IDENTIFICATION OF CLAIMS BEING CHALLENGED (§42.104(B))	13
VII. THE 720 PATENT AND ITS PROSECUTION HISTORY	14
A. The Underlying Technology	14
B. The 720 Patent.....	15
C. The 720 Patent’s Prosecution History	16
VIII. CLAIM CONSTRUCTION	16
IX. ORDINARY LEVEL OF SKILL IN THE ART	17
X. DESCRIPTION OF THE PRIOR ART	17
A. Cochran (Ex. 1005)	17

TABLE OF CONTENTS
(continued)

	Page
B. HyperSQL (Ex. 1006)	19
C. Zellweger (Ex. 1007).....	20
D. Microsoft Access (Ex. 1009).....	21
E. Janes (Ex. 1012)	22
F. Spiegel (Ex. 1013).....	23
XI. GROUND 1: COCHRAN IN VIEW OF ZELLWEGER RENDERS OBVIOUS CLAIMS 1-7, 15-19, 27-30	24
A. Claim-by-Claim Analysis.....	24
1. Claim 1	24
2. Claim 2	35
3. Claim 3	36
4. Claim 4	40
5. Claim 5	41
6. Claim 6	41
7. Claim 7	42
8. Claim 15	42
9. Claim 16	43
10. Claim 17	45
11. Claim 18	47
12. Claim 19	49
13. Claim 20	49
14. Claim 27	53
15. Claim 28	53
16. Claim 29	54
17. Claim 30	54
B. Motivation to Combine Cochran and Zellweger.....	55

TABLE OF CONTENTS
(continued)

	Page
XII. GROUND 2: HYPERSQL IN VIEW OF ZELLWEGER RENDERS OBVIOUS CLAIMS 1-5, 7, 15-19, 27-30	58
A. Claim-by-Claim Analysis.....	58
1. Claim 1	58
2. Claim 2	65
3. Claim 3	66
4. Claim 4	70
5. Claim 5	70
6. Claim 7	70
7. Claim 15	71
8. Claim 16	74
9. Claim 17	75
10. Claim 18	76
11. Claim 19	77
12. Claim 27	78
13. Claim 28	78
14. Claim 29	79
15. Claim 30	80
B. Motivation to Combine HyperSQL and Zellweger.....	80
XIII. GROUND 3: HYPERSQL IN VIEW OF ZELLWEGER AND MICROSOFT ACCESS RENDERS OBVIOUS CLAIMS 21-23	82
A. Claim-by-Claim Analysis.....	82
1. Claim 21	82
2. Claim 22	85
3. Claim 23	86
B. Motivation to Combine HyperSQL, Zellweger, and Microsoft Access.....	87

TABLE OF CONTENTS
(continued)

	Page
XIV. GROUND 4: HYPERSQL IN VIEW OF ZELLWEGER, MICROSOFT ACCESS AND JANES RENDERS OBVIOUS CLAIM 25	90
A. Claim-by-Claim Analysis.....	90
1. Claim 25	90
B. Motivation to Combine HyperSQL, Zellweger, Microsoft Access, and Janes	92
XV. GROUND 5: COCHRAN IN VIEW OF ZELLWEGER, JANES, AND SPIEGEL RENDERS OBVIOUS CLAIM 26	94
A. Claim-by-Claim Analysis.....	94
1. Claim 26	94
B. Motivation to Combine Cochran, Zellweger, Janes, and Spiegel.....	96
XVI. SECONDARY CONSIDERATIONS OF NON-OBVIOUSNESS	98
XVII. CONCLUSION.....	98
APPENDIX A – CLAIM LISTING	102

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GmbH</i> , IPR2019-01469, Paper 6 (Feb. 13, 2020).....	9
<i>Apple Inc. v. Achates Reference Publishing, Inc.</i> , IPR2013-00081, Paper 80 (PTAB June 2, 2014)	11
<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 15 (PTAB May 13, 2020)	6, 7, 8, 9
<i>Apple Inc. v. Maxell, Ltd.</i> , IPR2020-00204, Paper 11 (PTAB June 19, 2020)	8
<i>Applications in Internet Time, LLC v. RPX Corp.</i> , 897 F. 3d 1336 (Fed. Cir. 2018)	11
<i>Boston Scientific Corp. et al. v. Nevro Corp.</i> , IPR2020-01563, Paper 14 (PTAB Mar. 16, 2021)	9
<i>Ecobee Technologies ULC v. Causam Enterprises, Inc.</i> , IPR2022-01339, Paper 20 (PTAB Mar. 22, 2023)	11
<i>Expedia, Inc. et al v. International Business Machines Corp.</i> , Case No. IPR2018-01354, Paper 8, 16-20 (PTAB Jan. 15, 2019)	22
<i>In re Hall</i> , 781 F.2d 897 (Fed. Cir. 1986)	19, 22
<i>Oracle Corporation v. Vilox Technologies, LLC and Joseph DeBellis</i> , Case No. 1-23-cv-00126-MN (D. Del.).....	5
<i>Oticon Med. AB v. Cochlear Ltd.</i> , IPR2019-00975, Paper 15 (PTAB Oct. 16, 2019).....	7
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (<i>en banc</i>)	16
<i>Pure Storage, Inc. v. Realtime Data LLC</i> , Case No. IPR2018-00549, Paper 7 (July 23, 2018).....	9

Petition for *Inter Partes* Review of U.S. Patent No. 6,760,720

<i>Samsung Electronics Co., Ltd. v. Netlist, Inc.</i> , IPR2022-00615, Paper 62 (PTAB June 30, 2023)	11, 12
<i>Sand Revolution II, LLC v. Cont'l Intermodal Grp. – Trucking LLC</i> , IPR2019-01393, Paper 24 (PTAB June 16, 2020)	7, 8
<i>Synaptive Med. Inc. v. Karl Storz Endoscopy-Am., Inc.</i> , No. IPR2018-00462, 2018 WL 3469195 (PTAB July 16, 2018).....	9
<i>Uniloc 2017 LLC v. Facebook Inc.</i> , 989 F. 3d 1018 (Fed. Cir. 2021)	12
<i>Vilox Technologies, LLC and Joseph DeBellis v. Oracle Corporation</i> , Case No. 1-23-cv-00302-MN (D. Del.).....	4
<i>Vilox Technologies, LLC, et al. v. Oracle Corporation</i> , Case No. 6-22-cv-01254-FB (W.D. Tex.)	4
<i>Vilox Technologies, LLC, et al v. Salesforce, Inc.</i> , Case No. 6-22-cv-01255-FB (W.D. Tex.).....	5
<i>Vilox Technologies LLC v. Costco Wholesale Corporation</i> , Case No. 2-15-cv-02019-JRG-RSP (E.D. Tex.).....	5
<i>Vilox Technologies LLC v. Expedia, Inc. et al</i> , Case No. 2-15-cv-01457-JRG-RSP (E.D. Tex.).....	6
<i>Vilox Technologies LLC v. Express, Inc. et al</i> , Case No. 2-15-cv-02025 (E.D. Tex.).....	6
<i>Vilox Technologies LLC v. Foot Locker, Inc. et al</i> , Case No. 2-15-cv-01458-JRG-RSP (E.D. Tex.).....	6
<i>Vilox Technologies LLC v. L Brands, Inc. et al</i> , Case No. 2-15-cv-02028 (E.D. Tex.).....	5, 6
<i>Vilox Technologies LLC v. MindGeek USA Inc.</i> , Case No. 2-16-cv-01278-JRG-RSP (E.D. Tex.).....	5
<i>Vilox Technologies LLC v. Orbitz Worldwide, Inc. et al</i> , Case No. 2-15-cv-01459-JRG-RSP (E.D. Tex.).....	6

Petition for *Inter Partes* Review of U.S. Patent No. 6,760,720

<i>Vilox Technologies LLC v. The Priceline Group, Inc. et al</i> , Case No. 2-15-cv-01460-JRG-RSP (E.D. Tex.).....	6
---	---

<i>Vilox Technologies, LLC v. Salesforce, Inc.</i> , Case No. 3-23-cv-05047-AMO (N.D. Cal.)	5
--	---

<i>Western Digital Technologies, Inc. v. Ocean Semiconductor LLC</i> , IPR2021-00864, Paper 13 (PTAB Nov. 16, 2021).....	8
---	---

<i>WesternGeco LLC v. ION Geophysical Corp.</i> , 889 F.3d 1308 (Fed. Cir. 2018)	11
---	----

<i>Wi-Fi One, LLC v. Broadcom Corp.</i> , 887 F.3d 1329 (Fed. Cir. 2018)	11
---	----

Statutes

35 U.S.C. §§311-319.....	1
--------------------------	---

35 U.S.C. §315	4, 10, 12
----------------------	-----------

35 U.S.C. §325	9
----------------------	---

Other Authorities

37 C.F.R. §42	1, 3, 4
---------------------	---------

LISTING OF EXHIBITS

Exhibit	DESCRIPTION
1001	U.S. Patent No. 6,760,720 to DeBellis (the “720 Patent”)
1002	Patent Prosecution History for U.S. Patent No. 6,760,720 (the “720 FH”)
1003	Declaration of Dr. Erez Zadok (“Zadok”)
1004	Curriculum Vitae of Dr. Erez Zadok
1005	U.S. Patent No. 5,995,979 to Cochran, Filed Feb. 10, 1998 (“Cochran”)
1006	Mark Newsome et al., <i>HyperSQL: Web-based Query Interfaces for Biological Databases</i> , IEEE Computer Society, Proceedings of The Thirtieth Annual Hawaii International Conference on System Sciences (1997) (“HyperSQL”)
1007	U.S. Patent No. 6,433,797 to Zellweger, Filed July 15, 1999, Claiming Priority to Provisional Application No. 60/095,811, Filed Aug. 4, 1998 (“Zellweger”)
1008	<i>Reserved</i>
1009	<i>Using Microsoft Access 97</i> , Special Edition, by Roger Jennings (1997) (“Microsoft Access”)
1010	Library of Congress Catalog Entry for <i>Using Microsoft Access 97</i> , Special Edition, by Roger Jennings (1997) – Full Catalog Record
1011	Library of Congress Catalog Entry for <i>Using Microsoft Access 97</i> , Special Edition, by Roger Jennings (1997) – MARC Tags
1012	U.S. Patent No. 6,642,946 to Janes, Filed Aug. 12, 1999 (“Janes”)
1013	U.S. Patent No. 6,629,079 to Spiegel, Filed June 25, 1998 (“Spiegel”)

Exhibit	DESCRIPTION
1014	Declaration of Rachel J. Watters on Authentication of Publication: Mark Newsome et al., <i>HyperSQL: Web-based Query Interfaces for Biological Databases</i> , IEEE Computer Society, Proceedings of The Thirtieth Annual Hawaii International Conference on System Sciences (1997)
1015	Certification From the Library of Congress for <i>Using Microsoft Access 97</i> , Special Edition, by Roger Jennings (1997)
1016	Docket, <i>Vilox Technologies, LLC and Joseph DeBellis v. Oracle Corporation</i> , Case No. 1-23-cv-00302-MN (D. Del.)
1017	Docket, <i>Oracle Corporation v. Vilox Technologies, LLC and Joseph DeBellis</i> , Case No. 1-23-cv-00126-MN (D. Del.)
1018	Docket, <i>Vilox Technologies, LLC v. Salesforce, Inc.</i> , Case No. 3-23-cv-05047-AMO (N.D. Cal.)
1019	Plaintiffs' Amended Complaint For Patent Infringement (Corrected), <i>Vilox Technologies, LLC and Joseph DeBellis v. Oracle Corporation</i> , Case No. 1-23-cv-00302-MN (D. Del. Sept. 11, 2023)
1020	Plaintiff's Original Complaint For Patent Infringement, <i>Vilox Technologies, LLC, et al. v. Salesforce, Inc.</i> , Case No. 6-22-cv-01255-FB (W.D. Tex. Dec. 5, 2022)
1021	Plaintiff's Original Complaint For Patent Infringement, <i>Vilox Technologies, LLC et al. v. Oracle Corporation</i> , Case No. 6-22-cv-01254-FB (W.D. Tex. Dec. 5, 2022)
1022	Letter from Oracle to Costco regarding Costco Request for Indemnity, dated Jan. 20, 2016 [PROTECTIVE ORDER MATERIAL - FILED UNDER SEAL]
1023	Order Re Stipulation of Dismissal Without Prejudice, <i>Vilox Tech., LLC v. Costco Wholesale Corp.</i> , No. 2-15-cv-2019-JRG-RSP (E.D. Tex. Mar. 22, 2016)
1024	<i>Reserved</i>

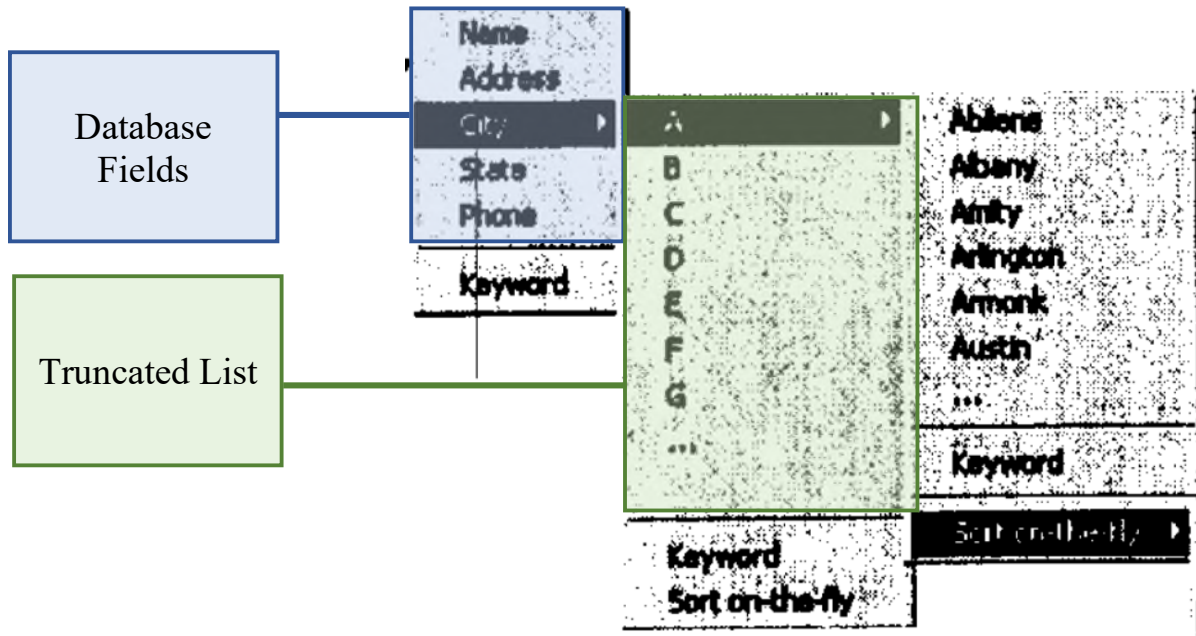
Exhibit	DESCRIPTION
1025	<i>Microsoft Computer Dictionary</i> , 4 th ed., Microsoft Press, 1999 (“Microsoft Dictionary”)
1026	<i>4.4BSD Programmer’s Reference Manual</i> , O’Reilly & Associates, Inc., 1994 (“BSD Programmer Manual”)
1027	<i>4.4BSD Programmer’s Supplementary Documents</i> , O’Reilly & Associates, Inc., 1994 (“BSD Supplementary Docs”)
1028	ANSI/ISO/IEC International Standard (IS), Database Language SQL, Part 2: Foundation (SQL/Foundation), International Standards Organization (ISO), 1999 (“SQL Standard”)
1029	<i>Oracle SQL*Plus: The Definitive Guide</i> , by Jonathan Gennick, O’Reilly & Associates, Inc., 1999 (“SQL*Plus Guide”)
1030	<i>ODBC 3.5 Developer’s Guide</i> , by Roger E. Sanders, McGraw-Hill, 1999 (“ODBC Guide”)
1031	<i>Oracle Developer, Guidelines for Building Applications, Release 6.0</i> , September 1998 (“Oracle Building Apps Guide”)

I. INTRODUCTION

Pursuant to 35 U.S.C. §§311-319 and 37 C.F.R. §42, Petitioner Oracle Corporation (“Petitioner”) respectfully requests *Inter Partes* Review (“IPR”) of Claims 1-7, 15-23, and 25-30 (the “Challenged Claims”) of U.S. Patent No. 6,760,720 (Ex. 1001, the “720 Patent”). There is more than a reasonable likelihood that Petitioner will prevail with respect to at least one claim.

The 720 Patent discloses and claims methods related to a database “search engine.” The purportedly novel method involves determining the database’s makeup and providing (1) the searchable fields (*i.e.*, the database column headings), and (2) the potential search terms for those fields (*i.e.*, the entries within each column). The potential search terms can be provided, for example, via a pull-down menu in the interface.

The method also requires “truncating” the list of search terms when the list is too voluminous. Specifically, if the number of terms/entries for a given data field exceeds a particular number, the patent requires that characters be removed from the list such that each entry is still represented but the list length is shortened. For example, the interface may truncate a list to the first letters of each entry (*i.e.*, A-Z) and the user can then select a letter to view all entries starting with that letter:



Ex. 1001, Fig. 10 (annotated).

None of these concepts is new. Search engines for databases were well known in the prior art, and most of them automatically determined relevant information about the database for the user. *See* Zadok, ¶¶119-122. This Petition relies on two such systems.

Truncated menus were also well known and obvious, and the prior art expressly teaches the use of such truncated lists for displaying voluminous lists from databases. *See* Zadok, ¶¶70-75, 93-118.

Additional limitations recited in dependent claims add nothing inventive, claiming well-known and obvious variations/features of database search engines having little or nothing to do with the claimed searching methods. Petitioner

requests that the Board institute trial and find the Challenged Claims unpatentable under §103.

II. REQUIREMENTS FOR PETITION FOR INTER PARTES REVIEW

A. Grounds for Standing (37 C.F.R. §42.104(a))

Petitioner certifies that the 720 Patent is available for IPR and Petitioner is not barred or estopped from requesting IPR of the Challenged Claims on the grounds identified herein.

B. Notice of Lead and Backup Counsel, Service Information, and Fee Information (37 C.F.R. §§42.8(b)(3-4), 41.10(a), 42.103)

Pursuant to 37 C.F.R. §§42.8(b)(3-4) and 42.10(a), Petitioner provides the following designation of Lead and Back-Up counsel:

Lead Counsel	Back-Up Counsel
K. Patrick Herman Registration No. 75,018 (P52PTABDocket@orrick.com) Postal & Hand-Delivery Address: Orrick, Herrington & Sutcliffe LLP 51 West 52nd Street New York, NY 10019 Tel: 212-506-3596	Jared Bobrow Jason Yu <i>Pro Hac Vice</i> to be submitted (PTABDocketJ3B3@orrick.com, J2YPTABDocket@orrick.com) Postal & Hand-Delivery Address: Orrick, Herrington & Sutcliffe LLP 1000 Marsh Road Menlo Park, CA 94025 Tel: 650-614-7400 Shane D. Anderson <i>Pro Hac Vice</i> to be submitted (PTABDocket9SA@orrick.com) Postal & Hand-Delivery Address: Orrick, Herrington & Sutcliffe LLP

	355 S. Grand Ave., Ste. 2700 Los Angeles, CA 90071 Tel: 213-629-2020
--	--

Petitioner consents to service by electronic mail at the addresses listed above.

Pursuant to 37 C.F.R. §42.10(b), Petitioner’s Power of Attorney is submitted.

The USPTO is authorized to charge the filing fee and any other fees incurred by Petitioner to the deposit account of Orrick, Herrington & Sutcliffe LLP: 15-0665.

C. Notice of Real-Parties-in-Interest (37 C.F.R. §42.8(b)(1))

Petitioner is the real-party-in-interest (“RPI”). No other parties exercised or could have exercised control over this Petition. No other parties funded or directed this Petition. *See* Office Patent Trial Practice Guide, 77 Fed. Reg. 48759-60. Petitioner does not dispute that the estoppel provisions apply to its wholly-owned subsidiaries. *See* 35 U.S.C. §315(e).

D. Notice of Related Matters (37 C.F.R. §42.8(b)(2))

According to USPTO assignment records, the 720 Patent is currently assigned to Joseph L. DeBellis (“DeBellis” or “Patent Owner”). The 720 Patent is at issue in several pending litigations involving Patent Owner and Vilox Technologies, LLC (“Vilox”), an entity solely owned by Patent Owner:

- *Vilox Technologies, LLC and Joseph DeBellis v. Oracle Corporation*, Case No. 1-23-cv-00302-MN (D. Del.), which was transferred from W.D. Tex. (*Vilox Technologies, LLC, et al. v. Oracle Corporation*,

Case No. 6-22-cv-01254-FB (W.D. Tex.)) (“The Oracle Litigation”).

DeBellis moved to intervene as a plaintiff and the court granted that motion.

- *Vilox Technologies, LLC v. Salesforce, Inc.*, Case No. 3-23-cv-05047-AMO (N.D. Cal.), which was transferred from W.D. Tex. (*Vilox Technologies, LLC, et al v. Salesforce, Inc.*, Case No. 6-22-cv-01255-FB (W.D. Tex.)) (“The Salesforce Litigation”). DeBellis moved to intervene as a plaintiff and the court has yet to issue an order on that motion.
- *Oracle Corporation v. Vilox Technologies, LLC and Joseph DeBellis*, Case No. 1-23-cv-00126-MN (D. Del.) (“The Oracle DJ Action”) (together with The Oracle Litigation and The Salesforce Litigation, “The Active Litigations”).

The 720 Patent was previously at issue in:

- *Vilox Technologies LLC v. MindGeek USA Inc.*, Case No. 2-16-cv-01278-JRG-RSP (E.D. Tex.).
- *Vilox Technologies LLC v. Costco Wholesale Corporation*, Case No. 2-15-cv-02019-JRG-RSP (E.D. Tex.) (consolidated with *Vilox Technologies LLC v. L Brands, Inc. et al*, Case No. 2-15-cv-02028 (E.D. Tex.)).

- *Vilox Technologies LLC v. Express, Inc. et al*, Case No. 2-15-cv-02025 (E.D. Tex.) (consolidated with *Vilox Technologies LLC v. L Brands, Inc. et al*, Case No. 2-15-cv-02028-JRG-RSP (E.D. Tex.)).
- *Vilox Technologies LLC v. Expedia, Inc. et al*, Case No. 2-15-cv-01457-JRG-RSP (E.D. Tex.) (consolidated with *Vilox Technologies LLC v. Foot Locker, Inc. et al*, Case No. 2-15-cv-01458-JRG-RSP (E.D. Tex.)).
- *Vilox Technologies LLC v. Orbitz Worldwide, Inc. et al*, Case No. 2-15-cv-01459-JRG-RSP (E.D. Tex.) (consolidated with *Vilox Technologies LLC v. Foot Locker, Inc. et al*, Case No. 2-15-cv-01458-JRG-RSP (E.D. Tex.)).
- *Vilox Technologies LLC v. The Priceline Group, Inc. et al*, Case No. 2-15-cv-01460-JRG-RSP (E.D. Tex.) (consolidated with *Vilox Technologies LLC v. Foot Locker, Inc. et al*, Case No. 2-15-cv-01458-JRG-RSP (E.D. Tex.)).

III. THE BOARD SHOULD NOT DENY INSTITUTION UNDER §314

The Board typically considers the factors set forth in *Apple v. Fintiv* in determining whether institution should be granted. IPR2020-00019, Paper 11, (PTAB Mar. 20, 2020) (precedential) (“*Fintiv*”). “[W]here a petition presents compelling evidence of unpatentability,” however, the PTAB should not apply the

Fintiv analysis. Interim Procedure for Discretionary Denials at 2 (June 21, 2022). Thus, “compelling, meritorious challenges will be allowed to proceed at the PTAB even where district court litigation is proceeding in parallel.” *Id.*, 4-5. This is the case here.

Even if examined, the *Fintiv* factors weigh strongly in favor of institution.

A. Existence Or Likelihood Of A Stay

This factor is neutral. Petitioner intends to file a motion to stay if the Board institutes IPR proceedings. Until the motion is adjudicated, any attempt to predict the outcome would require speculation. *See Sand Revolution II, LLC v. Cont’l Intermodal Grp. – Trucking LLC*, IPR2019-01393, Paper 24 at 7 (PTAB June 16, 2020) (informative) (“*Sand Revolution*”); *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 15 at 12 (PTAB May 13, 2020).

B. Proximity Of Trial Date To Final Written Decision

This factor favors institution. The Active Litigations are in their early stages and no court has issued a scheduling order. Exs. 1016-1018. That no trial date has been set weighs in favor of institution. *See Oticon Med. AB v. Cochlear Ltd.*, IPR2019-00975, Paper 15 at 23–24 (PTAB Oct. 16, 2019) (Institution Decision) (precedential as to §§II.B and II.C).

C. Investment In The Parallel Proceeding

This factor favors institution because the courts and parties will have invested little in the parallel actions. “If, at the time of the institution decision, the district

court has not issued orders related to the patent at issue in the petition, this fact weighs against exercising discretion to deny institution.” *Fintiv* at 10. Each Active Litigation is in its early stages and no court has issued any order related to the merits of the 720 Patent.

D. Overlapping Issues

This factor heavily favors institution because this Petition covers many claims that are not at issue in The Active Litigations. Specifically, of the 22 claims challenged in this Petition, only two (Claims 1 and 3) are asserted in the parallel litigations. *See* Exs. 1019-1020. Moreover, should the Board institute an IPR proceeding on the 720 Patent, Petitioner stipulates that it will not pursue any instituted grounds as invalidity defenses in district court, thus eliminating any overlap in issues. Such stipulations weigh in favor of institution. *See Sand Revolution* at 11-12; *Apple Inc. v. Maxell, Ltd.*, IPR2020-00204, Paper 11 at 15-17 (PTAB June 19, 2020).

E. Identity Of Parties

Petitioner and Patent Owner are parties in certain Active Litigations. However, IPRs are routinely instituted in such circumstances. *See, e.g., Western Digital Technologies, Inc. v. Ocean Semiconductor LLC*, IPR2021-00864, Paper 13 at 17, 37 (PTAB Nov. 16, 2021).

F. Other Circumstances Including The Merits

This factor favors institution because the merits of this Petition are strong. *See Fintiv* at 14-15. The Petition is premised on prior art combinations that the Patent Office did not consider. Even when a petition is not particularly strong, this factor is neutral where the petition merely “demonstrate[s] a reasonable likelihood” of success. *See Boston Scientific Corp. et al. v. Nevro Corp.*, IPR2020-01563, Paper 14 at 26-27 (PTAB Mar. 16, 2021).

IV. DENIAL UNDER 35 U.S.C. §325(D) WOULD BE INAPPROPRIATE

The Board should exercise its discretion to institute under §325(d) because the Petition relies on art and arguments that the Examiner did not consider. In applying §325(d), the Board uses a two-part framework: (1) determine whether the same or substantially the same art or arguments were previously presented to the Office; and (2) if so, determine whether the Petitioner has demonstrated that the Office erred in a manner material to patentability of the Challenged Claims. *Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6 at 8 (Feb. 13, 2020) (precedential).

Here, none of the prior art in the grounds of this Petition was cited or disclosed by the applicant or considered by the Examiner. *See Pure Storage, Inc. v. Realtime Data LLC*, Case No. IPR2018-00549, Paper 7 at 11 (July 23, 2018); *Synaptive Med.*

Inc. v. Karl Storz Endoscopy-Am., Inc., No. IPR2018-00462, 2018 WL 3469195, at *4 (PTAB July 16, 2018).

If Patent Owner argues that the Board should exercise its discretion to deny institution, Petitioner respectfully requests a reply brief.

V. THE BOARD SHOULD NOT DENY INSTITUTION UNDER 35 U.S.C. §315(B)

The first complaint asserting the 720 Patent against Petitioner (or one of Petitioner’s products) was served December 5, 2022. Ex. 1021. The Petition is timely under 35 U.S.C. §315(b).

Although Vilox sued Costco Wholesale Corporation (“Costco”)—an Oracle customer—on November 30, 2015 (the “Costco Litigation”), Costco is neither in privity with Petitioner with respect to the 720 Patent nor an RPI.

In Costco Litigation, Vilox claimed that Costco infringed four patents including the 720 Patent and U.S. Patent No. 7,188,100 (the “100 Patent”). Petitioner was not a party to that litigation. Petitioner agreed, at Costco’s request, to indemnify Costco as to the unrelated 100 Patent. However, Petitioner did not indemnify or defend Costco as to the 720 Patent. *See* Ex. 1022. Petitioner had no obligation to do so, because the 720 Patent claim did not target Petitioner’s products or conduct. *Id.* Petitioner had no control over Costco’s defense of the 720 Patent, did not fund any part of that defense, and had no interest in or ability to litigate the validity of the 720 Patent. *Id.* The PTAB has declined to find similarly situated

parties to be in privity, especially where the petitioner did not, in fact, defend or indemnify. *See, e.g., Wi-Fi One, LLC v. Broadcom Corp.*, 887 F.3d 1329, 1340-41 (Fed. Cir. 2018); *WesternGeco LLC v. ION Geophysical Corp.*, 889 F.3d 1308, 1321-22 (Fed. Cir. 2018); *Samsung Electronics Co., Ltd. v. Netlist, Inc.*, IPR2022-00615, Paper 62 at 23-26 (PTAB June 30, 2023); *Apple Inc. v. Achates Reference Publishing, Inc.*, IPR2013-00081, Paper 80 at 10-12 (PTAB June 2, 2014).

Moreover, the privity analysis hinges on the parties' relationship as to the specific patent in question. *See WesternGeco*, 889 F.3d at 1318-19. That Petitioner indemnified Costco with respect to other patents does not make the Costco Litigation relevant. *See generally id.* That Petitioner was separately sued years later confirms that Costco and Petitioner have distinct, unrelated interests regarding the 720 Patent. This additionally weighs in favor of finding no privity. *See, e.g., Ecobee Technologies ULC v. Causam Enterprises, Inc.*, IPR2022-01339, Paper 20 at 11 (PTAB Mar. 22, 2023).

Costco also is not an RPI. The determination of who is an RPI "is a highly fact-dependent question." *Applications in Internet Time, LLC v. RPX Corp.*, 897 F.3d 1336, 1342 (Fed. Cir. 2018) ("*AIT*"); Consolidated Trial Practice Guide, 13 (citing *Taylor v. Sturgell*, 553 U.S. 880 (2008)). Considerations include whether a party "is a clear beneficiary that has a preexisting, established relationship with the petitioner," *AIT*, 897 F.3d, 1351, whether a party "exercises [or could exercise]

control over a petitioner’s participation in a proceeding, or whether a []party is funding the proceeding or directing the proceeding.” *Uniloc 2017 LLC v. Facebook Inc.*, 989 F. 3d 1018, 1028 (Fed. Cir. 2021) (alterations in original) (citing *AIT*, 897 F.3d at 1342–43). None of these considerations apply to Costco. Costco has at best a customer relationship with Petitioner. This alone, however, does not make it an RPI. *See, e.g., Samsung Elecs. Co. v. Netlist, Inc.*, IPR2022-00615, Paper 62, at 13 (PTAB June 30, 2023) (a “standard customer-manufacturer relationship” is “insufficient to create a time bar”). The parties have no preexisting, established relationship as to 720 Patent or the subject matter of this Petition. No product sold by Costco is or can be at issue in the Active Litigations. Exs. 1019-1020. Costco is not indemnifying Petitioner (nor is Petitioner indemnifying Costco). Costco played no role whatsoever in the preparation or funding of this Petition. Costco does not stand to benefit from this Petition because it settled the claims against it years ago. Ex. 1023. The 720 Patent expired in 2020. *See* Ex. 1001, Cover. Thus, any financial obligations Costco may have had with respect to the patent must have ended long ago.

Accordingly, the Board should not deny institution under 35 U.S.C. §315(b).

VI. IDENTIFICATION OF CLAIMS BEING CHALLENGED (§42.104(B))

Petitioner requests IPR of claims 1-7, 15-23, and 25-30 of the 720 Patent. This Petition explains why the claims are unpatentable, supported by the accompanying Declaration of Dr. Erez Zadok (Ex. 1003, “Zadok”).

Petitioner relies on the following references:

- (1) U.S. Patent No. 5,995,979 (“Cochran”)
- (2) Mark Newsome et al., *HyperSQL: Web-based Query Interfaces for Biological Databases*, IEEE Computer Society, Proceedings of The Thirtieth Annual Hawaii International Conference on System Sciences (1997) (“HyperSQL”)
- (3) U.S. Patent No. 6,433,797 (“Zellweger”)
- (4) *Using Microsoft Access 97*, Special Edition, by Roger Jennings (1997) (“Microsoft Access”)
- (5) U.S. Patent No. 6,642,946 (“Janes”)
- (6) U.S. Patent No. 6,629,079 (“Spiegel”)

Each reference is prior art. *See* §X. The application underlying the 720 Patent was filed Feb. 25, 2000, and does not claim priority to any earlier application. Ex. 1001, Cover. In this Petition, it is assumed that the 720 Patent is entitled to a priority date of Feb. 25, 2000.

Petitioner challenges the Challenged Claims as follows:

Ground	Claim(s)	Statutory Grounds ¹
1	1-7, 15-20, 27-30	Obvious over Cochran in view of Zellweger
2	1-5, 7, 15-19, 27-30	Obvious over HyperSQL in view of Zellweger
3	21-23	Obvious over HyperSQL in view of Zellweger and Microsoft Access
4	25	Obvious over HyperSQL in view of Zellweger, Microsoft Access, and Janes
5	26	Obvious over Cochran in view of Zellweger, Janes, and Spiegel

VII. THE 720 PATENT AND ITS PROSECUTION HISTORY

A. The Underlying Technology

The 720 Patent relates to accessing and searching databases. Ex. 1001, Title, Abstract, 1:7-9. “[D]atabases require a consistent structure, termed a schema, to organize and manage the information.” *Id.*, 1:38-65. Access to the database relies on sending commands to the database in a data sublanguage such as “Structured English Query Language (SQL).” *Id.*

Before the earliest priority date for the 720 Patent, databases, SQL, various methods to search databases, and various technologies used to access databases were well known. Zadok, ¶¶64-129.

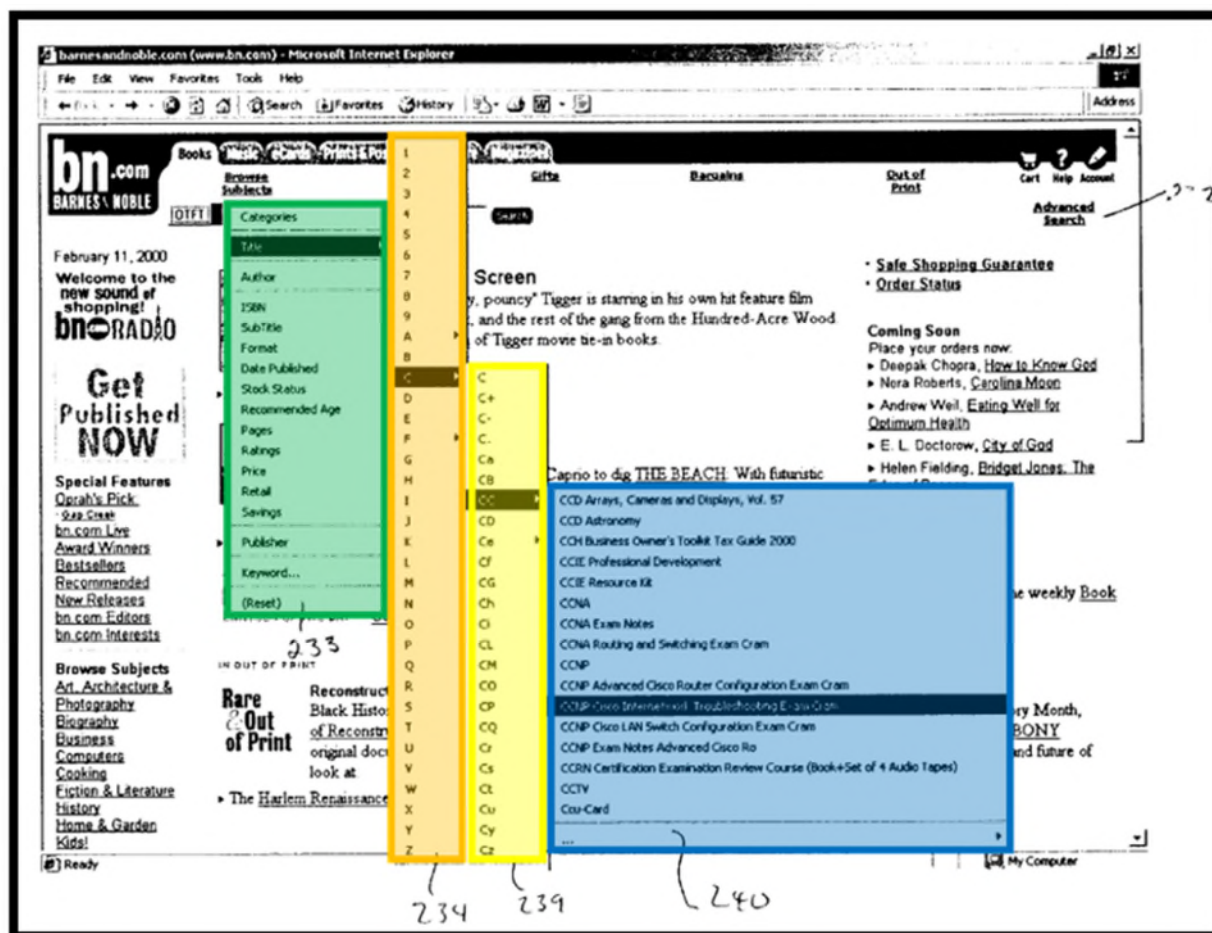
¹ Petitioner also relies on the knowledge of a POSITA for all grounds.

B. The 720 Patent

The 720 Patent is directed to “allowing a user to access data in [a] database without having to know anything about the database structure.” Ex. 1001, 2:30-35. The claimed methods involve determining the searchable categories or “fields” within a database (*i.e.*, column headings) and providing the user with each potential search term (*i.e.*, entries within the columns). *See, e.g., id.*, 5:42-58. The fields and search terms may be provided via a graphical user interface (“GUI”) with “pulldown” menus. *Id.*, 2:55-59.

The list of search terms is “truncated” into a representative set of data when the list is voluminous, *i.e.*, when the list exceeds a specified number of entries. *See id.*, Claims 1, 3, 5:63-67, 2:25-28.

Figure 13, for example, discloses searching a database of books using the claimed methods. The searchable categories or “fields” (in green) include Categories, Title, Author, ISBN, Format, Recommended Age, etc. The user selects the field “title” and, because the number of titles is voluminous, the list of titles is truncated into a representative list (orange). That representative list represents each title by the first character in the title (A, B, etc.). The user navigates using this truncated list.



Id., Fig. 13 (annotated).

C. The 720 Patent's Prosecution History

The application that resulted in the 720 Patent was rejected three times. The examiner never considered the prior art presented in this Petition. *See* Ex. 1002, pp. 120-135, 159-173, 198-213; *see also* Zadok, ¶¶138-142.

VIII. CLAIM CONSTRUCTION

The Board applies the same *Phillips* claim construction standard that is used in federal district court. *See Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005)

(*en banc*). For purposes of this Petition, the Challenged Claims' terms can be afforded their plain and ordinary meaning. *See* Zadok, ¶¶144-145.

IX. ORDINARY LEVEL OF SKILL IN THE ART

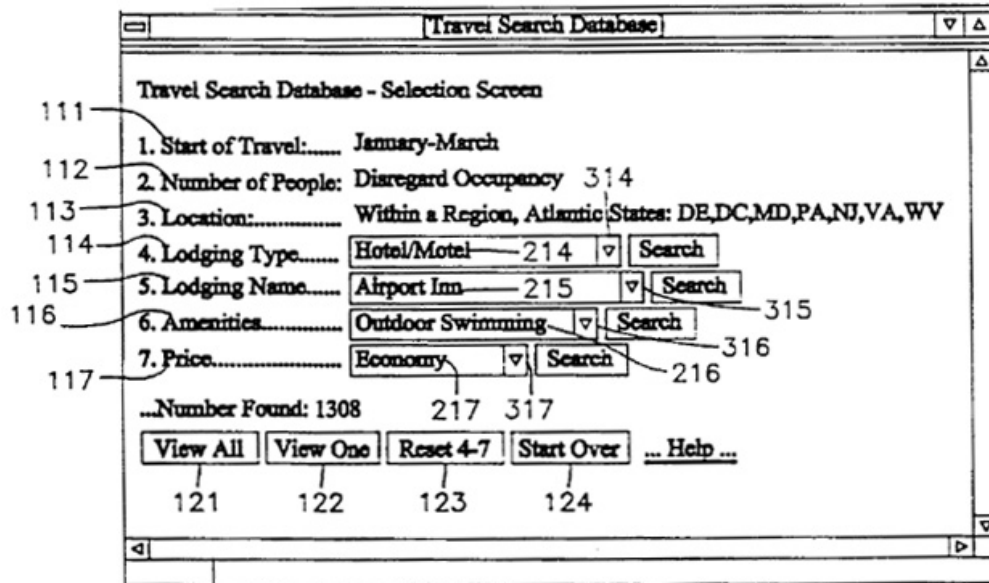
A person of ordinary skill in the art at the time of the alleged invention (“POSITA”) would have had at least a bachelor’s degree in computer science, electrical engineering, or a related subject, and at least two years of experience working with databases, computer networks, and related web technologies. A more advanced degree can substitute for less work experience and vice versa. Zadok, ¶¶57-61.

X. DESCRIPTION OF THE PRIOR ART

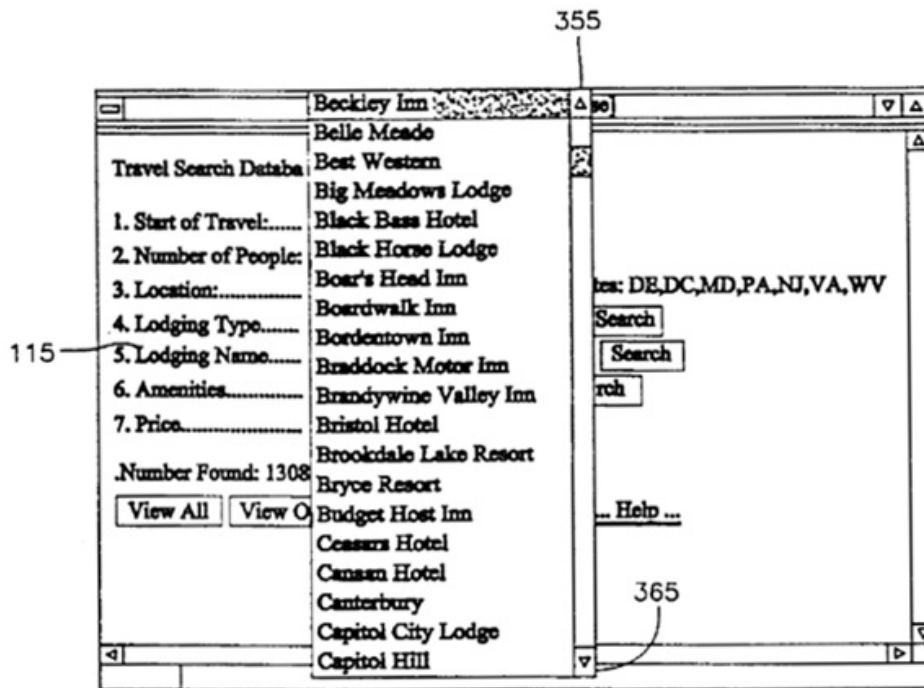
A. Cochran (Ex. 1005)

Cochran is a patent filed February 10, 1998. It is prior art under at least §102(e).

Cochran discloses a method for accessing a database using a GUI displaying a plurality of fields and available search terms. Cochran, Abstract, 1:15-19. Search terms are displayed and selected in “pick boxes.” *Id.*, 10:13-19.



Id., Fig. 4b.

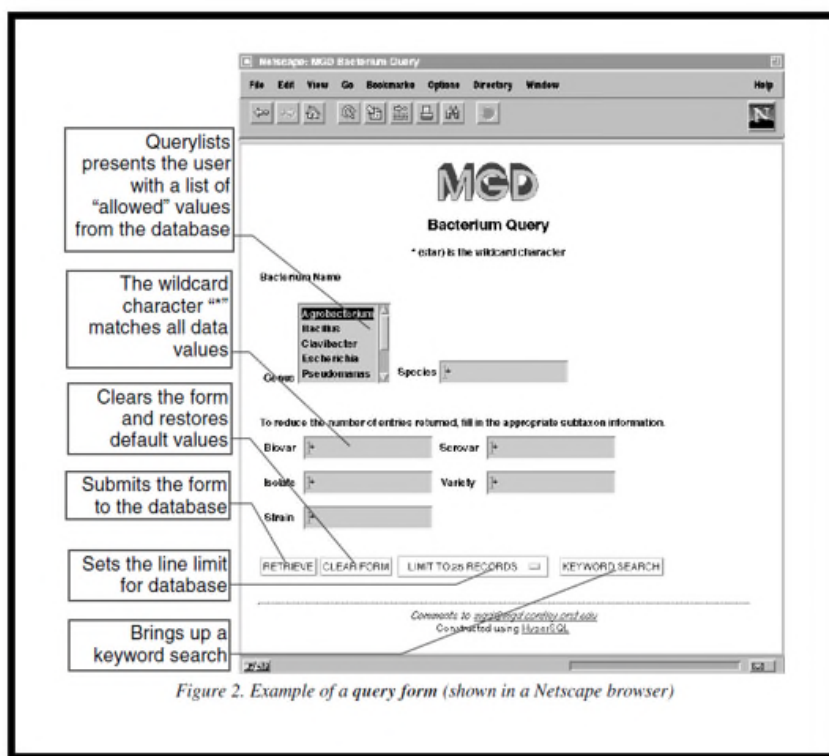


Id., Fig. 4e. These figures show seven data fields. In Figure 4e, the user expanded the Lodging Name “pick box,” which lists the available entries for that field. *Id.*, 12:26-13:67; *see also id.*, 4:29-50, 8:54-9:22.

B. HyperSQL (Ex. 1006)

HyperSQL is a conference paper dated January 1997. It was published, catalogued, and available at libraries within a few days or at most two to three weeks after April 22, 1998. *See* Ex. 1014. Thus, HyperSQL is prior art under at least §102(b). *See In re Hall*, 781 F.2d 897, 899 (Fed. Cir. 1986).

HyperSQL is a browser-based interface for searching complex databases. HyperSQL, 329. HyperSQL discloses a website with “query forms” that allow users to specify search criteria for database data fields. *Id.*, 329-333, Figs. 2-5. Search terms can be selected from “querylists,” which are scrollable lists of “allowed” values from the database. *Id.*, 333, Fig. 2.



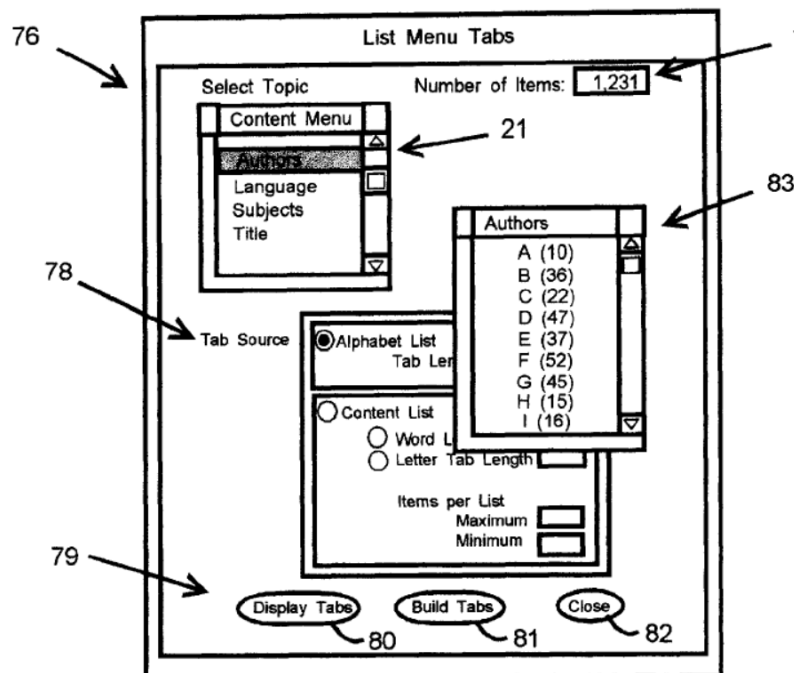
Id., Fig. 2. Figure 2 shows available data fields such as “Genus” and “Species.”

C. Zellweger (Ex. 1007)

Zellweger is a patent filed July 15, 1999. It is prior art under §102(e).

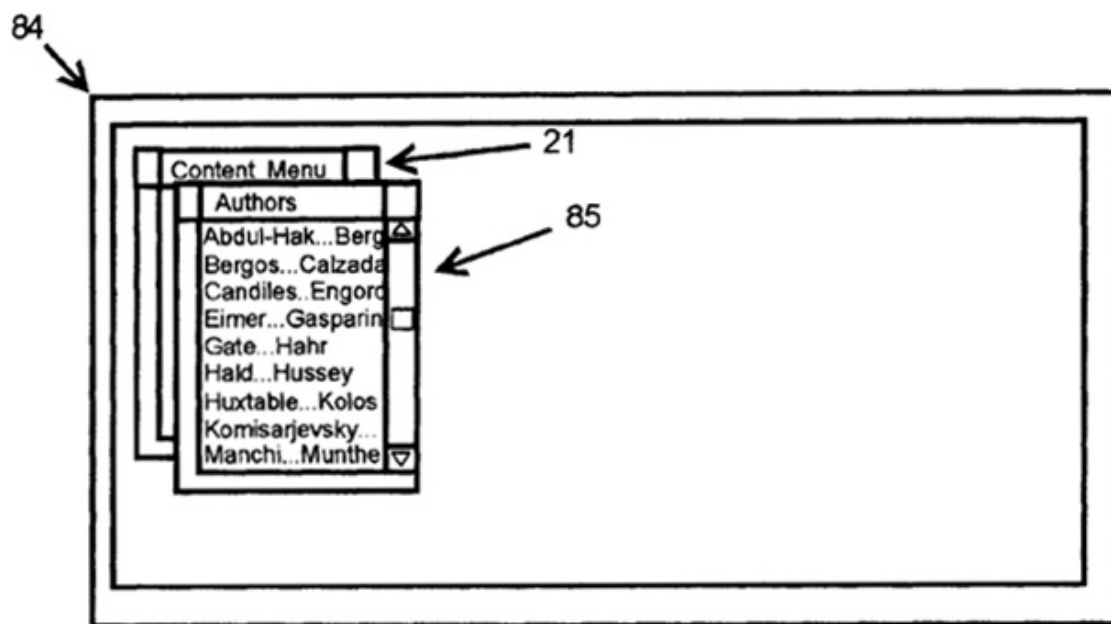
Zellweger discloses a method for reducing the size of voluminous lists contained in menus like those disclosed in Cochran and HyperSQL. Zellweger, Abstract, 2:30-43. The method involves shortening such voluminous lists into navigable “Alphabet Lists” or “Content Lists.” *Id.*, 4:21-52. Some users of Zellweger referred to as “menu developers” or “developers” may customize menu parameters. *See id.*, Abstract, 2:30-59, 4:7-52; Zadok, ¶¶157-162.

Alphabet Lists reduce the volume of the original list by truncating the items down to a defined number of leading letters, shown in Figure 6b. Zellweger, 4:21-52.



Id., Fig. 6b.

Content Lists organize the entries into intervals, such as the interval of author last names in Figure 7. *Id.*, 4:26-31, 4:45-53.



Id., Fig. 7.

D. Microsoft Access (Ex. 1009)

Using Microsoft Access 97 is a textbook published and sold by Que Corporation in 1997 that contains detailed information on how to use Microsoft Access 97. *See* Microsoft Access; *id.*, Copyright Page. A copy of Microsoft Access (Library of Congress Catalog No. 96-71443) was cataloged by—with call number QA76.9 .D3 J474 1997, Copy 1—and available from the Library of Congress by January 10, 1997. Ex. 1015. Microsoft Access was cataloged under “Microsoft Access” and “Database management” subjects and assigned Dewey class no. 005.75/65. Exs. 1010-1011. Microsoft Access is prior art under at least §102(b).

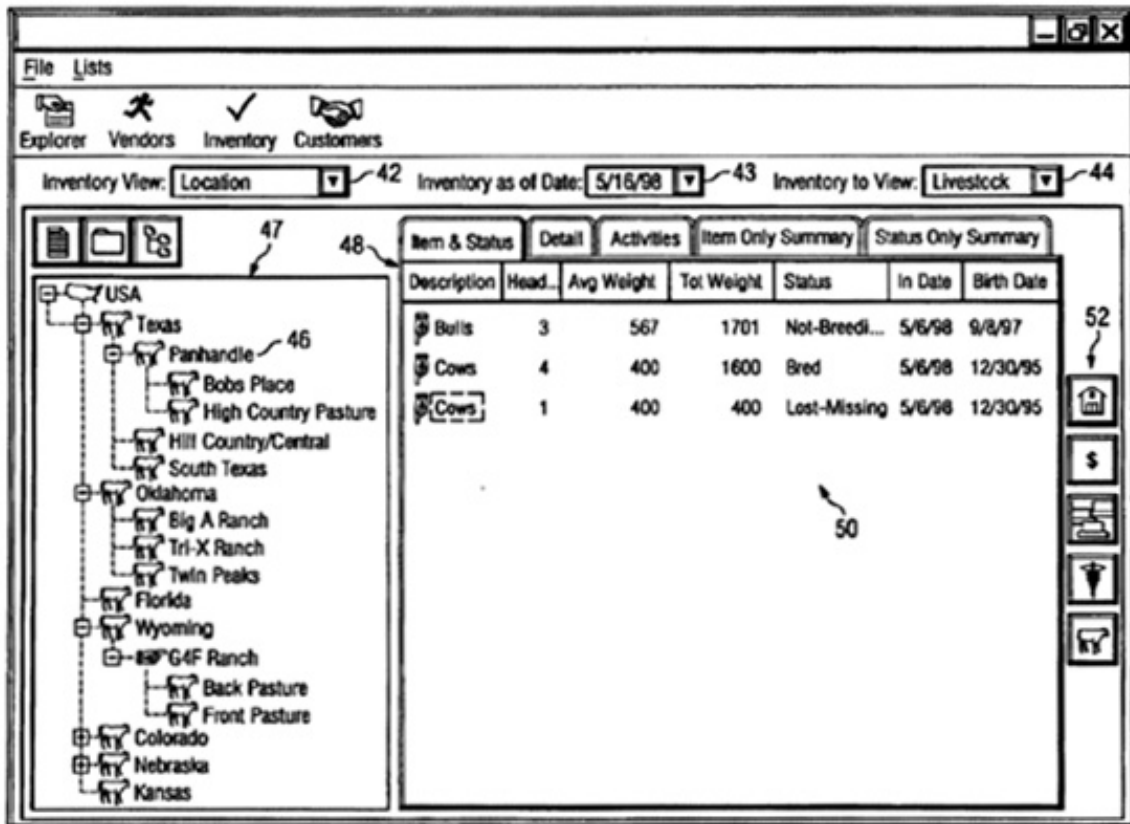
See In re Hall, 781 F.2d at 899; *Expedia, Inc. et al v. International Business Machines Corp.*, Case No. IPR2018-01354, Paper 8, 16-20 (PTAB Jan. 15, 2019) (finding textbook deposited and indexed at the Library of Congress publicly accessible).

Microsoft Access 97 was a popular relational database management system used to create desktop and client/server database applications. Microsoft Access, 1. Microsoft Access discloses many well-known operations related to the usage of databases, including combining search results and using logical operators. *Id.*, 291.

E. Janes (Ex. 1012)

Janes is a patent filed August 12, 1999. It is prior art under §102(e).

Janes discloses using a GUI to manage data in a database. Janes, Title, Abstract, 2:10-62. Janes teaches using a hierarchical menu system with “nodes” to organize the data. *Id.*, 2:10-62, 4:30-5:60. When a user clicks on a node, the GUI displays the associated data as icons and text. *Id.* Janes also teaches that entities in the GUI may be dragged and dropped onto other entities to complete actions. *Id.*, 2:48-62, 5:51-60, 11:22-12:44.



Id., Fig. 2A.

F. Spiegel (Ex. 1013)

Spiegel is a patent filed June 25, 1998. Spiegel is prior art under §102(e).

Spiegel discloses a visual e-commerce system that uses shopping carts with associated billing and shipping information. Spiegel, Abstract, 2:61-3:40. A user may add items for purchase graphically by dragging-and-dropping them onto shopping cart icons on the GUI. *Id.*, 5:39-6:16, 6:48-58, 10:31-54, Figs. 1, 3.

XI. GROUND 1: COCHRAN IN VIEW OF ZELLWEGER RENDERS OBVIOUS CLAIMS 1-7, 15-19, 27-30²

A. Claim-by-Claim Analysis

1. Claim 1

a. [1.PRE]

“A method implemented on a computer for searching databases, comprising:”

Cochran discloses “[a] method for managing the presentation of search terms on a computer terminal ... so that large computer databases can be searched efficiently.” Cochran, Abstract; *see also id.*, Figs. 7b, 3a, 4a, 4b, 4d, 1:30-37, 4:10-12; Zadok, ¶¶176-177.

b. [1.A]

“determining a database schema for a database, wherein the database includes database fields;”

Cochran discloses “a method for accessing records from a computer database.” Cochran, 1:15-16. The “[i]nformation is stored in the database as a collection of records, each record having multiple fields (also referred to as indexes or list identifiers) containing the data or information” (“*the database includes database fields*”).³ *Id.*, 1:21-29.

² Zadok, ¶¶173-317.

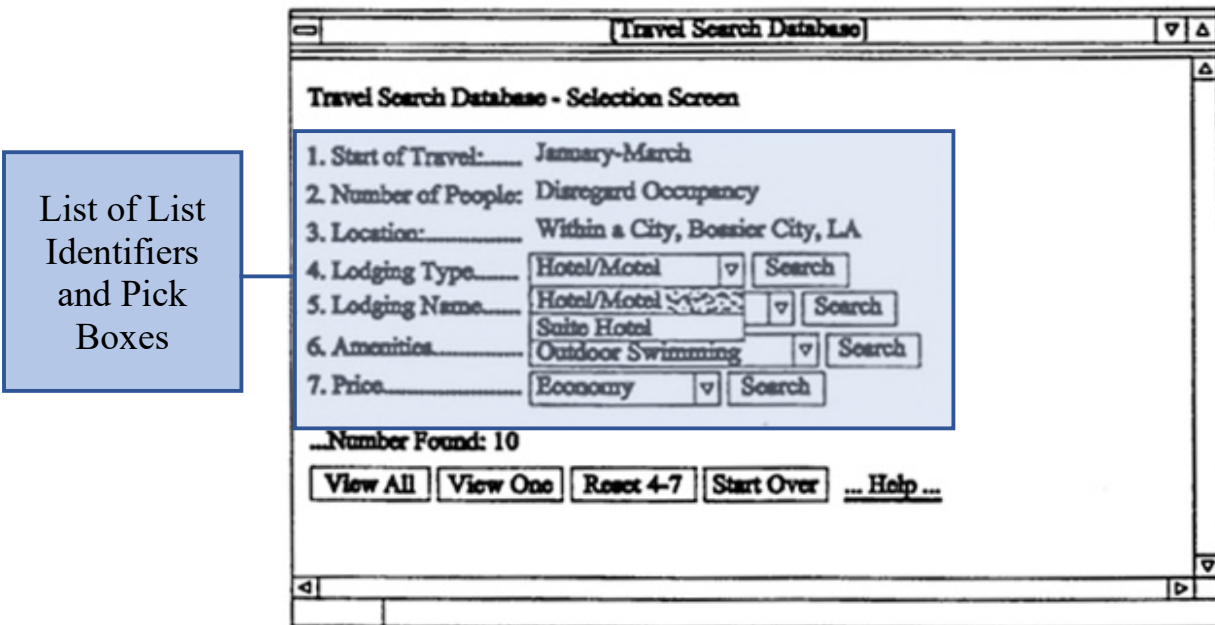
³ Italicized and quoted language is from the claims.

Cochran “display[s] a list of all available choices” because otherwise it “would be difficult for the user to know how [the] database was organized without the visual presentation” of the data available in the database. Cochran, 11:38-48. A POSITA would have understood that this means that Cochran “*determine[s] a database schema for a database.*” Otherwise, Cochran would be unable to learn (and list) “all available choices” from the database. Zadok, ¶¶178-181. A POSITA would have understood that Cochran determines the database schema and then presents the user with choices dictated by that schema. *Id.*; Cochran, 7:20-27.

c. [1.B]

“providing a list of the database fields, wherein the list includes a descriptor indicating a data category;”

To facilitate database searching, Cochran provides a list of “list identifiers” and “pick boxes” (“*providing a list of database fields*”) on a GUI where the “list identifiers” “identify[] the various categories of information” available in the database to be searched (“*the list includes a descriptor indicating a data category*”). Zadok, ¶¶182-184; Cochran, 1:21-29, 7:19-47.

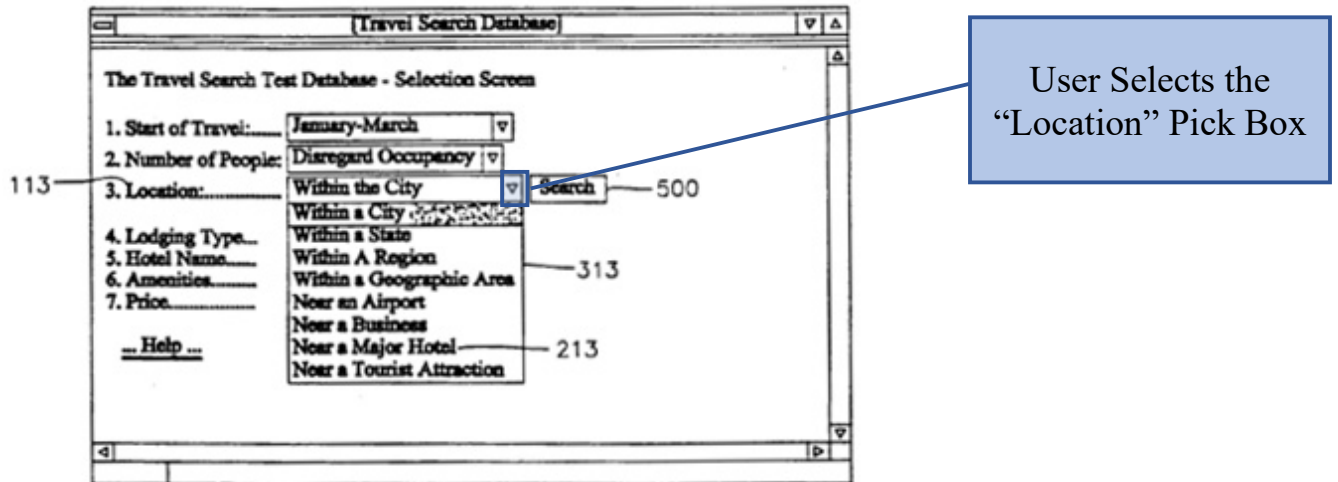


Id., Figs. 7b (annotated) (listing fields/descriptors for “Location,” “Lodging Type,” “Lodging Name,” etc.), 3a-4b, 4d-9d, Abstract, 7:13-47, 11:22-28.

d. [1.C]

“receiving a search selection for a database field on the provided list of the database fields;”

Cochran discloses that each listed database field/category includes an associated pick box that the user can click (“receiving a search selection for a database field” from “the provided list of database fields”), and the selected pick box expands to display “search terms” related to the associated database field. Cochran, 9:60-10:4. By determining which pick box has been selected (and thus needs to be populated with search terms for display) Cochran has received a search selection as required. Zadok, ¶¶185-187; *see also* Cochran, 1:30-47, 7:9-19, 6:31-39, 8:45-47.

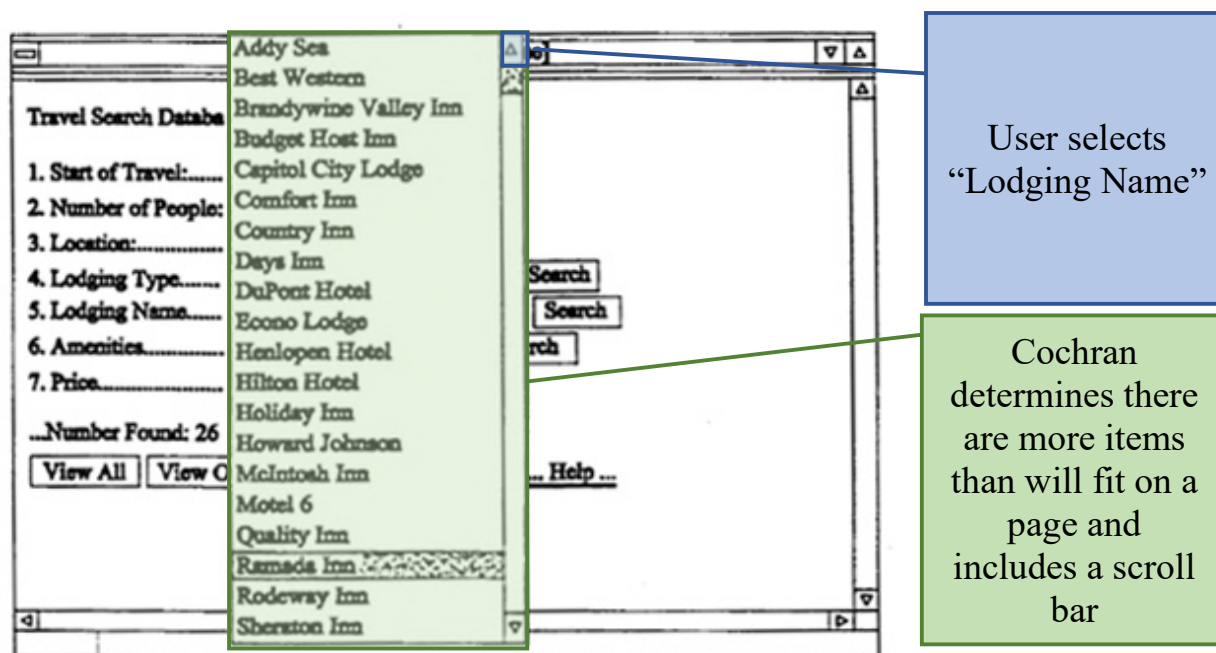


Id., Fig. 3d (annotated); *see also id.*, Figs. 3a-3c, 4a-4b, 4d-9d.

e. [1.D]

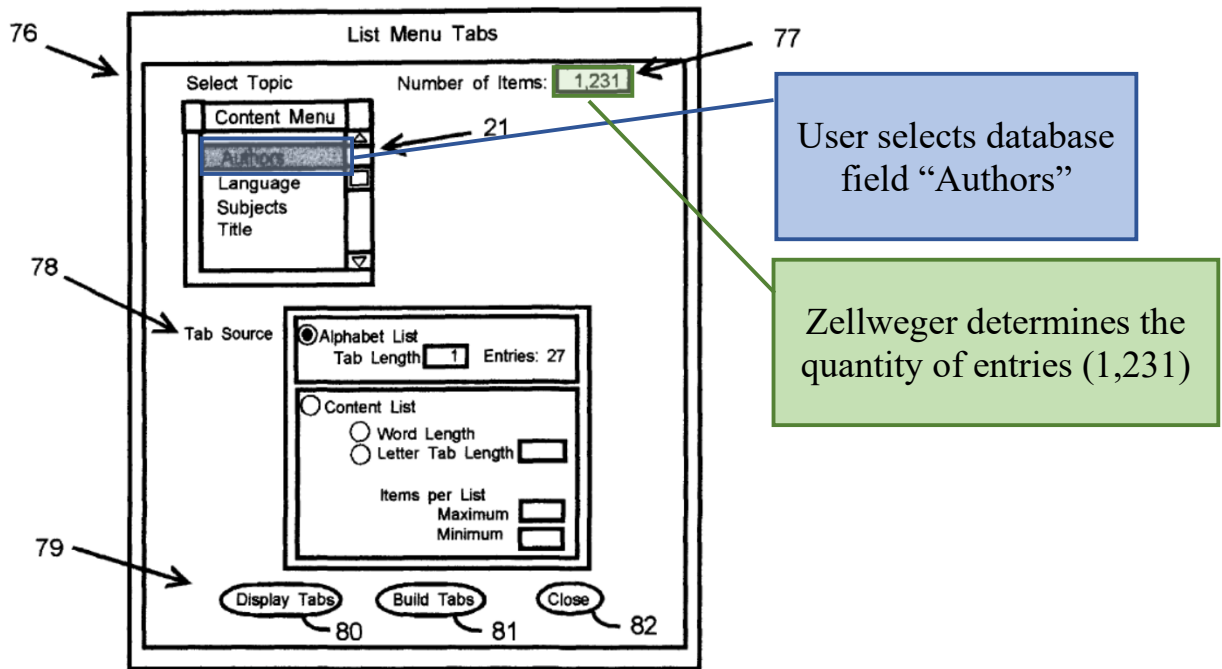
“determining a quantity of entries in the selected database field;”

Cochran determines the number of “search terms” (“*determining a quantity of entries*”) to display in an expanded pick box (“*in the selected database field*”) when Cochran generates the display of the expanded pick box. Zadok, ¶¶188-193, 201.



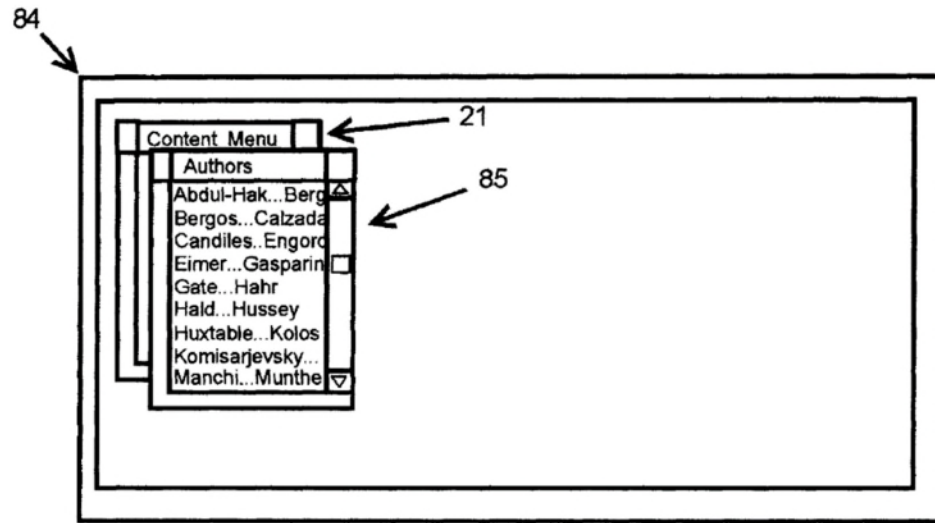
Cochran, Fig. 5d (annotated). Figure 5d shows that Cochran determines that the quantity of entries for “Lodging Name” (“*the selected database field*”) exceeds the number displayable simultaneously and adds a scroll bar. *Id.*, 10:57-65, Fig. 5d; Zadok, ¶¶188-189. Where the quantity of entries can be displayed simultaneously, Cochran does not display a scroll bar. *See* Cochran, Figs. 3-9. A POSITA would have understood that Cochran determines the quantity of entries in the database field to determine the screen space required by the list and thus whether a scroll bar is necessary. Zadok, ¶¶190-193.

To the extent not already taught by Cochran, this element would have been obvious in view of Zellweger. Zellweger discloses determining a number of items (“*quantity of entries*”) associated with a “*selected database field*” in several ways. Zadok, ¶¶194-201.

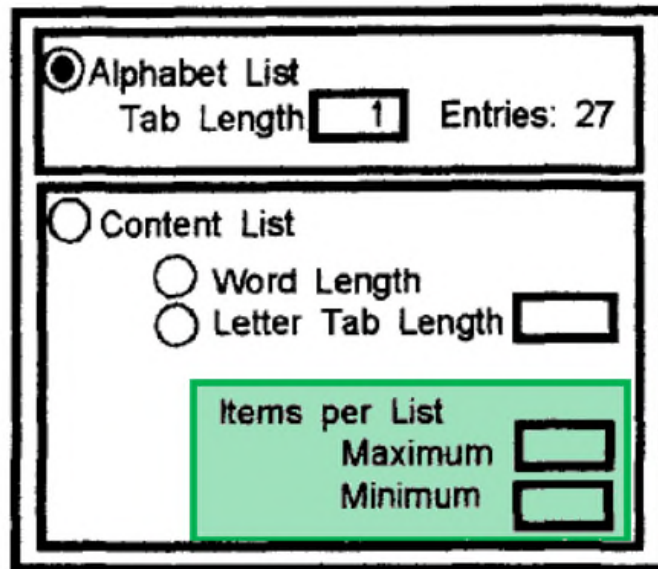


Zellweger, Fig. 6a (annotated). When the user selects a database field, Zellweger determines how many entries are in that field. *Id.*, 2:30-35, 4:7-16 (explaining that “the number of items” in a list is determined and “displayed in field 77”). Figure 6 shows a user selecting “Authors” and Zellweger determining that there are 1,231 entries in Authors. Zadok, ¶¶195-196.

Additionally, Zellweger discloses this limitation when generating a “Content List.” Zadok, ¶¶197-199. Zellweger can present an “Alphabet List” or “Content List.” Zellweger, 4:17-35. The Content List divides the list into intervals:



Id., Fig. 7. Figure 7 shows intervals listed “using the author’s last name as the first and last elements.” *Id.*, 4:45-53. Clicking on an entry opens a newly generated “new list menu that starts with” the first entry and ends with the last entry in the selected interval. *Id.*; Zadok, ¶¶198-199.



Zellweger, Fig. 6a (annotated). A user can set a maximum and a minimum number of items per list for the Content List. *Id.*, 4:26-31, 2:30-43. A POSITA would have

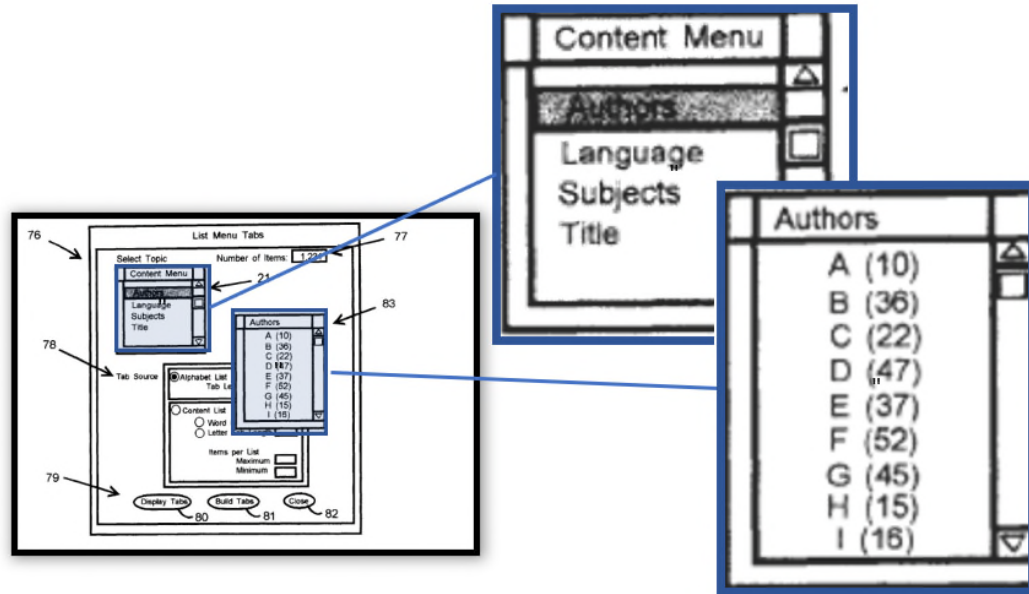
understood that Zellweger must determine the quantity of entries to determine if the minimum or maximum is satisfied. Zadok, ¶¶199-201.

f. [1.E]

“if the quantity exceed a specified amount, truncating data, and displaying the truncated data wherein the truncating reduces characters in one or more entries in the selected database field and the truncated data represents each of the entries in the selected database field; and”

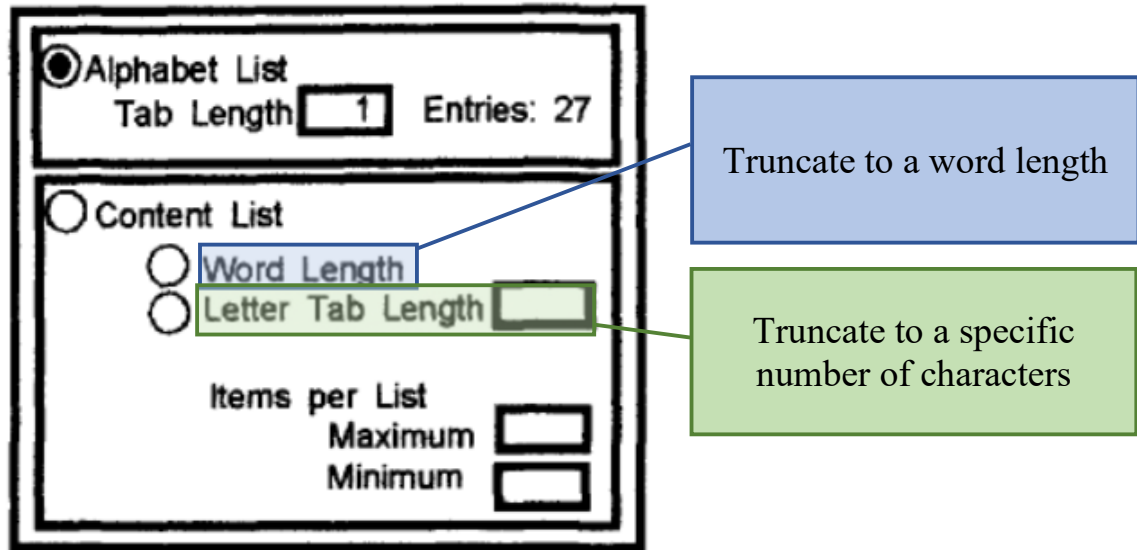
Cochran acknowledges that displaying “more [search terms] ... than can fit on the display screen simultaneously” necessitates the use of a scroll bar with “scroll buttons” to “scroll[] to display additional search terms.” Cochran, 13:58-14:7; *see also id.*, 10:48-65, 14:45-49, 15:4-11, 16:15-21. Thus, Cochran already contemplates adjusting its display based on the quantity of results. Zadok, ¶203.

Zellweger discloses methods for displaying such voluminous lists. Specifically, Zellweger discloses “*truncating*” the text of entries in a list by “*reduc[ing] characters in one or more entries of the selected database field*” and displaying them grouped into intervals using “Alphabet Lists” and “Content Lists” (“*displaying the truncated data*”). *See* §XI.A.1.e; Zadok, ¶¶204-212. An Alphabet List reduces the entries in the list to a user-defined number of letters and organizes entries into groups based on these letters. For example, Figure 6b shows truncating “Authors” entries to one letter:

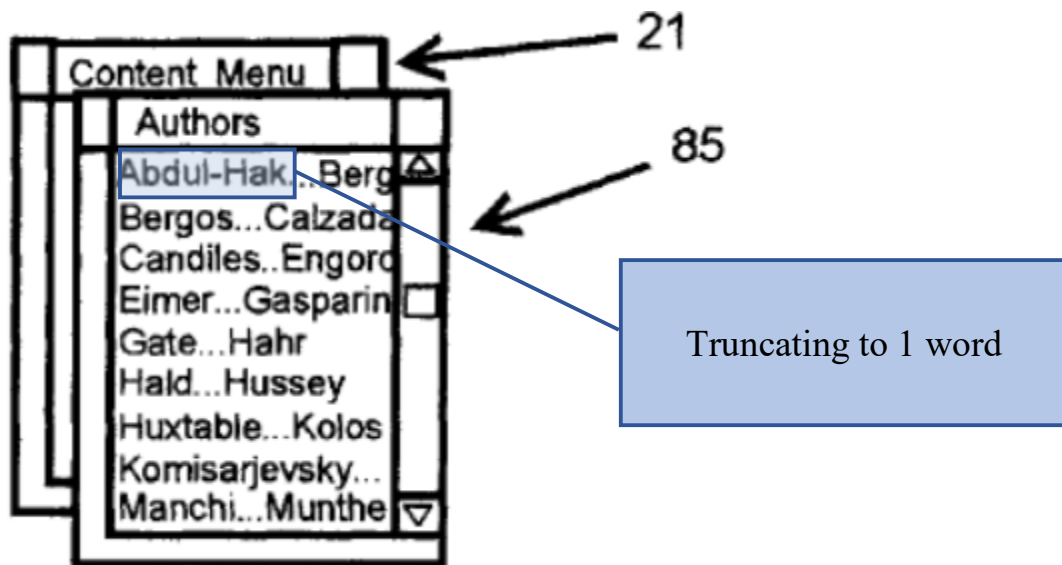


Zellweger, Fig. 6b (annotated), 4:36-44. Zellweger determined and reports in parentheses the quantity of database items within each truncated category. *See id.*

Alternatively, Zellweger may display entries in a Content List by truncating entries down to a word or a number of leading characters and organizing the entries into a set of intervals:



Id., Fig. 6a (annotated). Again, the user can also specify the number of items to be displayed “per List.” *See id.*



Id., Fig. 7 (annotated). Here, the entries for the “Authors” database field are truncated to one word length and organized into intervals. *Id.*, 4:45-53; Zadok, ¶¶204-208. The truncated data in either the Alphabet List or Content List

“represents” each of the authors (“entries”) in the “Authors” field (“selected database field”). *Id.*, ¶209.

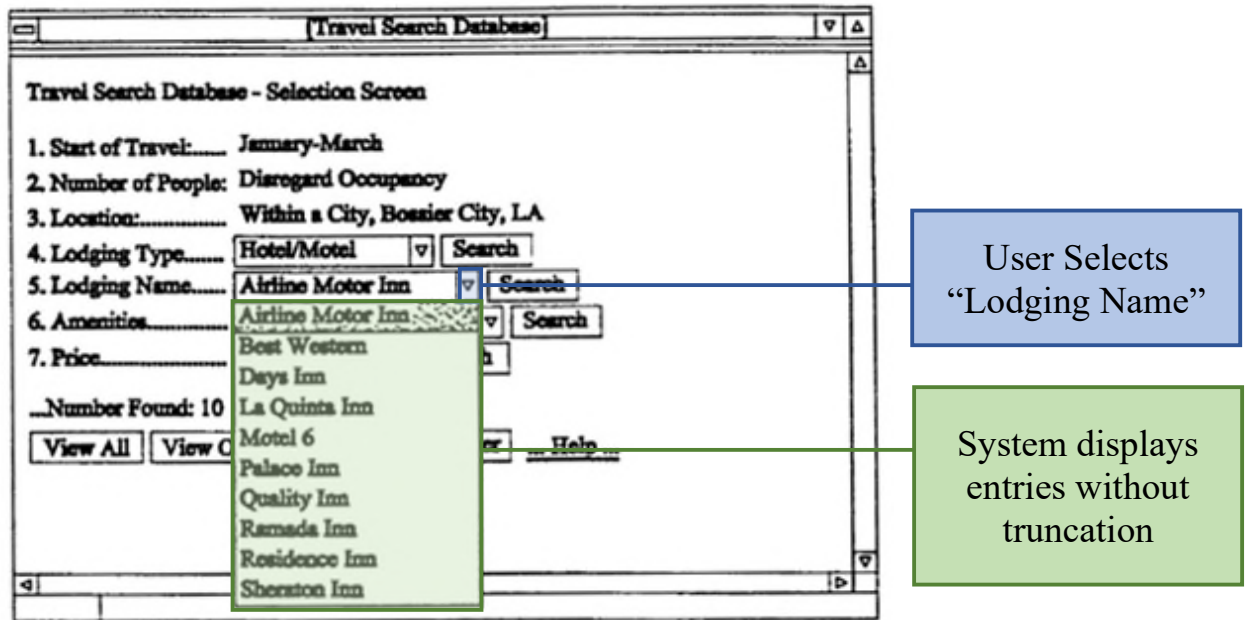
A POSITA would have considered it obvious to apply Zellweger’s truncation techniques to Cochran. Cochran teaches that its system determines pick box list size and adjusts displays if the list is too large. Zellweger teaches that a user can specify the minimum and maximum number of items displayed “per List.” Zellweger, 4:26-31. Zellweger also supplies other obvious ways besides scroll bars to display large lists of data, including truncating the list (and entries therein) using an Alphabet List or Content List. A POSITA would have considered it obvious to employ any of these techniques to achieve any user-specified list size (*e.g.*, 10, 50, 100). Zadok, ¶¶210-212. Thus, the references together teach truncating “*if the quantity exceed a specified amount.*” Zellweger, 4:26-35, Figs. 6a, 7; *see* §XI.A.1.e.

g. [1.F]

“if the quantity does not exceed the specified amount, displaying contents of the database field.”

Cochran in view of Zellweger discloses or renders obvious truncating the list of entries if the list exceeds a specified number of entries. §XI.A.1.f. Cochran in view of Zellweger would truncate when the number of entries exceeds the number of lines available on a display screen or based on some predetermined number selected by the user. *Id.* Accordingly, a POSITA would have understood or found obvious that the combined system displays the list without truncation (“displaying

contents of the database field”) when the list does not exceed that specified amount (“*if the quantity does not exceed the specified amount*”). *Id.*; Zadok, ¶¶213-217.



Cochran, Fig. 7c (annotated) (displaying entries for “Lodging Name” without truncation/modification because the list does not exceed the number of lines displayable on the screen), 11:22-48, 15:64-66, 16:15-21.

2. Claim 2

“The method of claim 1, further comprising providing a key word search.”

Cochran teaches that the database “may be searched by specifying the desired information sought as a ‘search term.’” Cochran, 1:30-37, 2:25-28, 9:18-22. A POSITA would have understood that selecting Cochran’s search terms in the pick boxes constitutes providing “key words” for searching the database. *See id.*, 10:13-18; Zadok, ¶218.

3. Claim 3

a. [3.PRE]

“A method implemented on a computer for searching a database, comprising:”

Cochran in view of Zellweger discloses this limitation. *See* §XI.A.1.a (Claim [1.PRE]); Zadok, ¶219.

b. [3.A]

“generating a list of data fields;”

Cochran in view of Zellweger discloses this limitation. *See* §XI.A.1.c (Claim [1.B]); Zadok, ¶220.

c. [3.B]

“receiving a first data field selection from the list of data fields;”

Cochran in view of Zellweger discloses this limitation. *See* §XI.A.1.d (Claim [1.C]). While they use different language, Claims 1 and 3 require the same thing:

- “search selection” (Claim 1) and “first data field selection” (Claim 3);
and
- “for a database field on the provided list of the database fields” (Claim 1) and “from the list of data fields” (Claim 3).

Cochran discloses these limitations for the same reasons. Zadok, ¶¶221-222.

d. [3.C]

“determining a first quantity indicative of a number of entries of the selected data field;”

Cochran in view of Zellweger discloses this limitation. *See* §XI.A.1.e (Claim [1.D]). While they use different language, Claims 1 and 3 require the same thing:

- “quantity of entries” (Claim 1) and “first quantity indicative of a number of entries” (Claim 3); and
- “in the selected database field” (Claim 1) and “of the selected data field” (Claim 3).

Cochran and Zellweger disclose these limitations for the same reasons. Zadok, ¶¶223-224.

e. [3.D]

“if the first quantity exceeds a specified limit, reducing a size of data to be displayed from the selected data field wherein the reducing reduces characters in one or more entries in the selected data field and the size-reduced data represents each of the entries in the selected field; and”

Cochran in view of Zellweger discloses this limitation. *See* §§XI.A.1.f (Claim [1.E]), XI.A.3.f (Claim [3.E]). While they use different language, Claims 1 and 3 require the same thing:

- “quantity” (Claim 1) and “first quantity” (Claim 3);
- “specified amount” (Claim 1) and “specified limit” (Claim 3);
- “truncating data” (Claim 1) and “reducing a size of data” (Claim 3);

- “the truncating” (Claim 1) and “the reducing” (Claim 3);
- “database field” (Claim 1) and “data field” (Claim 3);
- “truncated data” (Claim 1) and “size-reduced data” (Claim 3); and
- “selected database field” (Claim 1) and “selected field.” (Claim 3).

Cochran and Zellweger disclose these limitations for the same reasons. Zadok, ¶¶225-226.

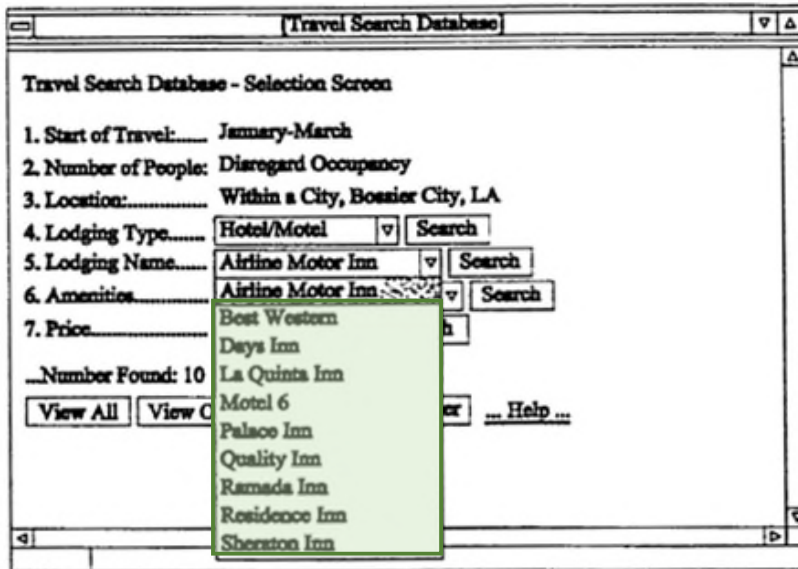
Claim 3 contains two terms that do not have antecedent basis: (1) “selected field” and (2) “date.” We make the following assumptions: (1) “selected field” refers to “the selected data field” and (2) “date” is read as “data.” Zadok, ¶227.

f. [3.E]

“displaying data from the selected data field.”

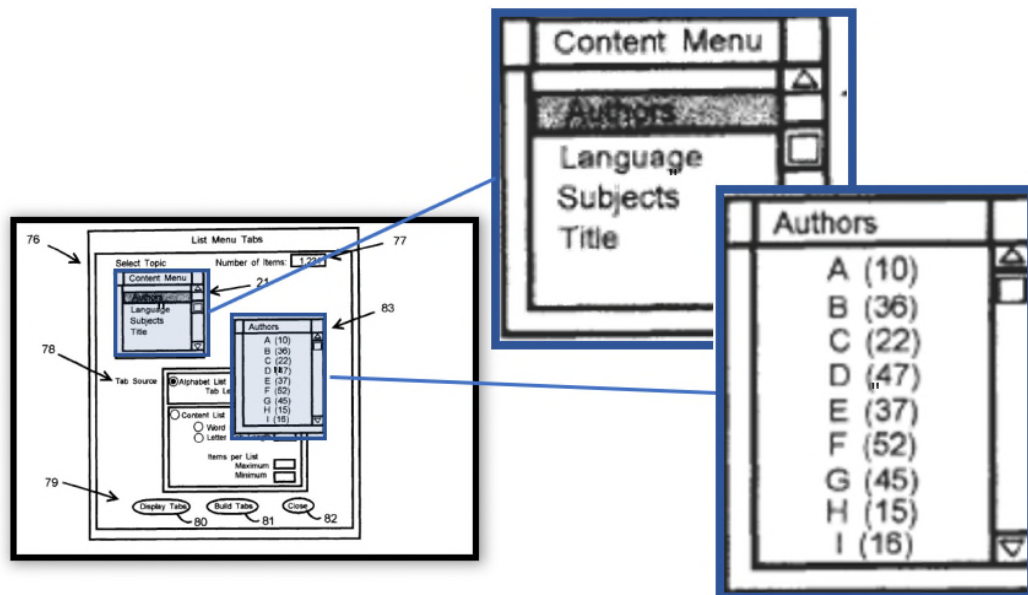
Cochran in view of Zellweger discloses this limitation. §§XI.A.1.e (Claim [1.D]) and XI.A.1.f (Claim [1.E]) show how the combined system “*display[s] data from the selected data field.*” Zadok, ¶¶228-234.

Cochran discloses displaying search terms (“*displaying data*”) in a user-selected pick box (“*from the selected data field*”):

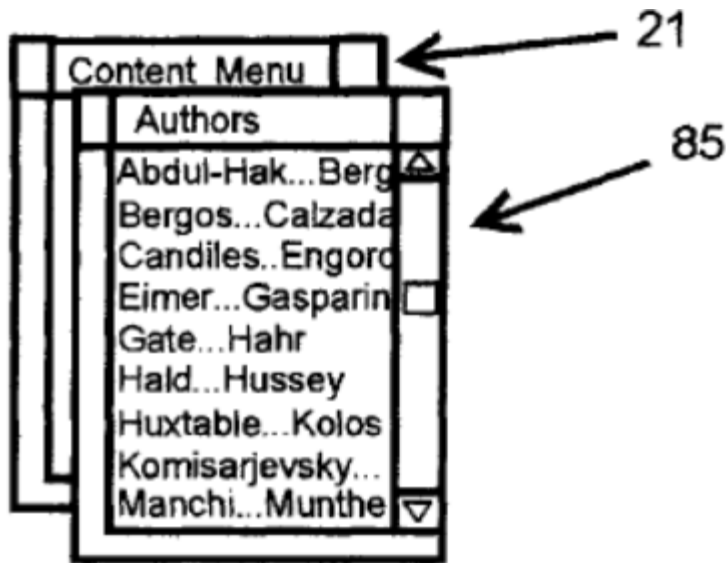


Cochran, Fig. 7c (annotated) (“Lodging Name” as the “*selected data field*”).

Zellweger discloses displaying the entries (“*displaying data*”) organized in either an Alphabet List or Content List for the selected topic (“*from the selected data field*”):



Zellweger, Fig. 6b (annotated) (“Authors” as the “*selected data field*”), 3:66-4:26.



Id., Fig. 7 (“Authors” as the “*selected data field*”), 4:26-31.

4. Claim 4

“The method of claim 3, wherein the specified limit is fixed.”

Zellweger splits a list of entries into intervals using “Content Lists,” for which a user may specify the minimum and maximum number of items per list. Zellweger, Abstract, 2:30-43, 4:26-52. Zellweger shows the “Minimum” and “Maximum” settings are plain text inputs. *Id.*, Fig. 6a. A POSITA would have understood that a user may input any number into the fields. Zadok, ¶¶235-236. The specified limit is “fixed” when the user sets the minimum items per list to one value and does not change it. *Id.*, ¶237.

5. Claim 5

“The method of claim 3, wherein the specified limit is variable.”

As explained in §XI.A.4 regarding Claim 4, Zellweger allows a user to set a minimum and maximum number of items per list. Zellweger describes a variable limit because the minimum value may be changed at will and thus when the value is changed it is “variable.” Zadok, ¶¶238-239. Additionally, as noted above, Cochran teaches that the specified limit may be based on the size of the screen, which is variable. See §XI.A.1.f; Zadok, ¶239.

6. Claim 6

“The method of claim 3, wherein the data are displayed on a terminal, and wherein the specified limit is determined dynamically, based on a characteristic of the terminal.”

Cochran discloses modifying the list (*e.g.*, with a scroll bar) if the number of entries exceeds that which will fit on a display screen. §§XI.A.1.e-XI.A.1.f. What fits on the screen is shown in the list (“*data are displayed on a terminal*”) and the rest are accessible using the scroll bar to show hidden entries. A POSITA would have understood that Cochran determines how many items fit on the screen simultaneously, which depends on the terminal’s display capabilities (“*wherein the specified limit is determined dynamically, based on a characteristic of the terminal*”). Zadok, ¶¶240-244.

7. Claim 7

“The method of claim 3, wherein the specified limit is a user-determined limit.”

As shown regarding Claims 4 and 5 (*see* §§XI.A.4, XI.A.5), Zellweger discloses that a user “can specify the maximum and minimum” value for the Content List “items per list,” Zellweger, 4:26-31, and thus they are “user-determined.” Zadok, ¶245.

8. Claim 15

a. [15.A]

“The method of claim 3, further comprising: receiving a first constraint, wherein the first constrain[t] is related to a data element in a data field; and”

Cochran discloses that a user selects a search term (“*receiving a first constraint*”). *See* §§XI.A.1.d and XI.A.3.c. A POSITA would have understood that the selected search term is a “constraint” because each term narrows the resulting search. Cochran, Abstract, 8:45-65; Zadok, ¶¶246-247. When a user presses the View All button, Cochran “locate[s]” and “presents the entire subset of records” relating to the selection(s). Cochran, 12:66-13:14.

Each of the search terms is “*related to*” a corresponding “*data element in a data field.*” For example, the price search terms are related to the Price field and so forth for each list item. *See, e.g.*, Cochran 4:2-9, 4:29-35, 12:44-65, 14:13-15:16, Figs. 4b, 4d-4g, 5b-5f; Zadok, ¶¶248-249.

b. [15.B]

“receiving one or more subsequent constraints, wherein search results are generated based on a combination of the first and the one or more subsequent constraints.”

Cochran receives “*one or more subsequent constraints*” when a user selects multiple additional search terms (“*constraints*”). Zadok, ¶250; See §XI.A.8.a. For example, a user may first select a search term “Within a City, Boston, MA” for the list identifier “Location” and *subsequently* select “Economy” as the search term for the list identifier “Price” and other additional constraints (*e.g.*, “Amenities”) as desired. Cochran, 16:9-17:13.

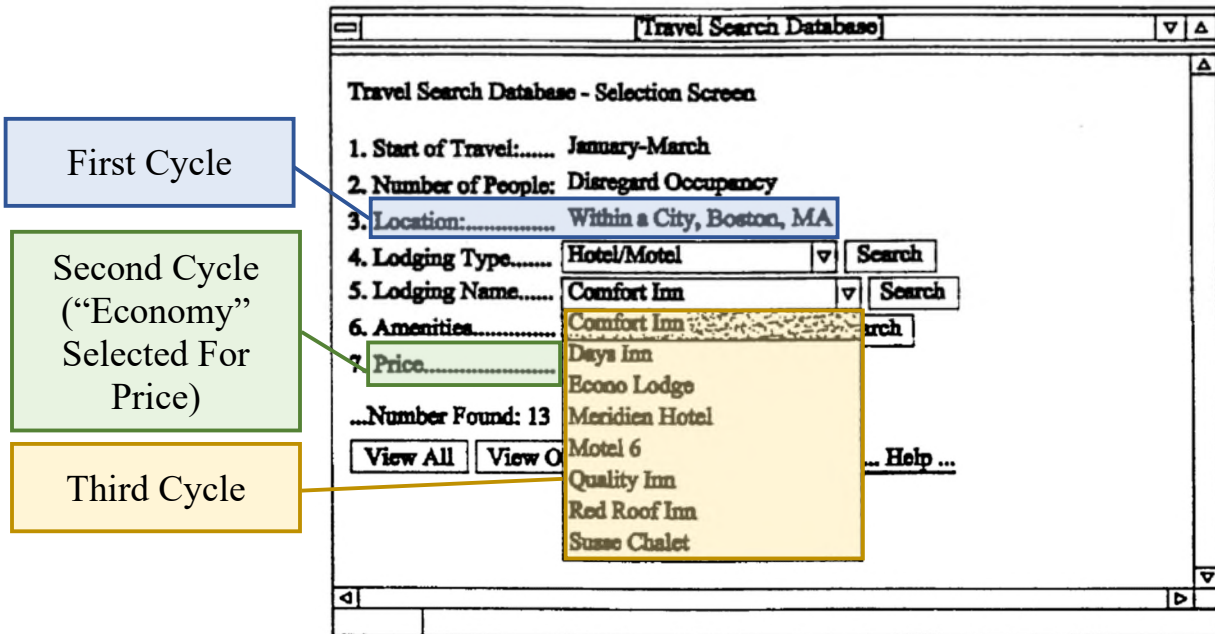
Results are “*generated based on a combination of the first*” and “*subsequent constraints*” in two ways. First, the user’s selected search terms may be applied in combination to arrive at a narrowed list of options in the other pick boxes (*e.g.*, the “Lodging Type”). Cochran, Figs. 8a-9d, 16:38-17:13. Second, Cochran discloses that the user may “activat[e] the View All button 121” which “presents the entire subset of records to the user,” *id.*, 12:66-13:3 and applies all selected search terms (“*constraints*”) to the search. *Id.*; Zadok, ¶¶251-254.

9. Claim 16

“The method [of] claim 3, wherein each display of data comprises a search cycle, and further comprising receiving a refresh command, the refresh command resetting all search cycles.”

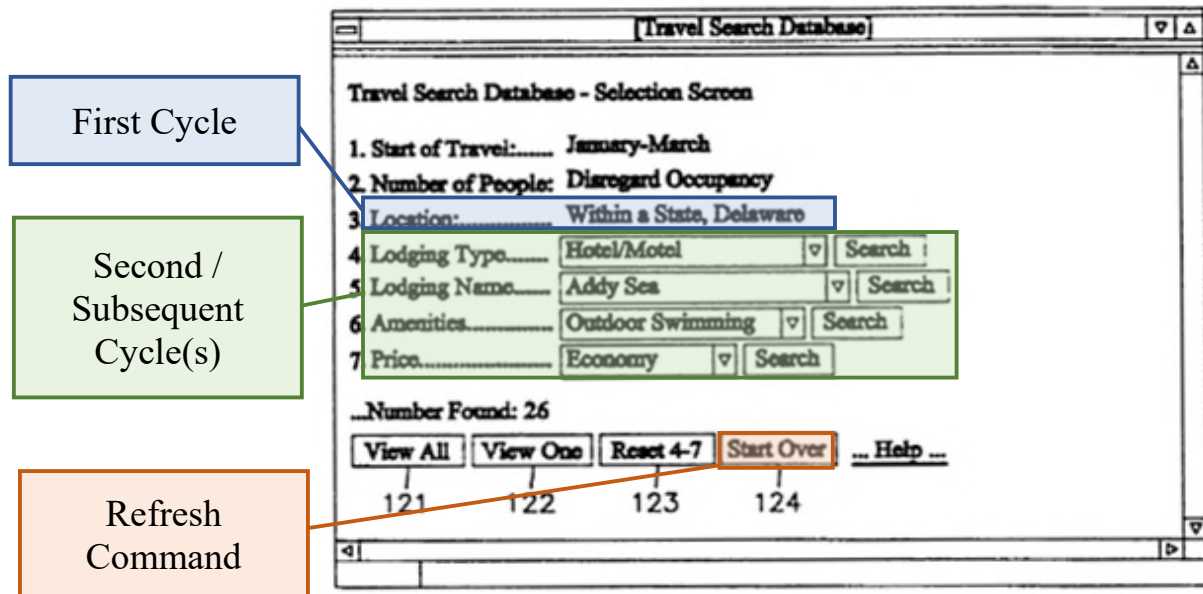
Cochran allows a user to select several search terms, where each selection causes the other pick box search terms to be updated (“*each display of data*

comprises a search cycle”). See Claim 15 (§§XI.A.8.a, XI.A.8.b); Cochran, 8:45-65 (explaining that an initial selection is used to “create updated lists for the list identifiers that were not originally searched”); Zadok, ¶¶255-265. One example of multiple “*search cycles*” is shown below:



Cochran, Fig. 9c (annotated) (three search cycles), 14:13-43, 12:66-13:13, 16:39-46, Abstract.

A user can issue a command by clicking the “Start Over” button that resets all pick box search terms selections (“*receiving a refresh command, the refresh command resetting all search cycles*”). Cochran, 13:19-26, Fig. 5b; Zadok, ¶¶266-269.



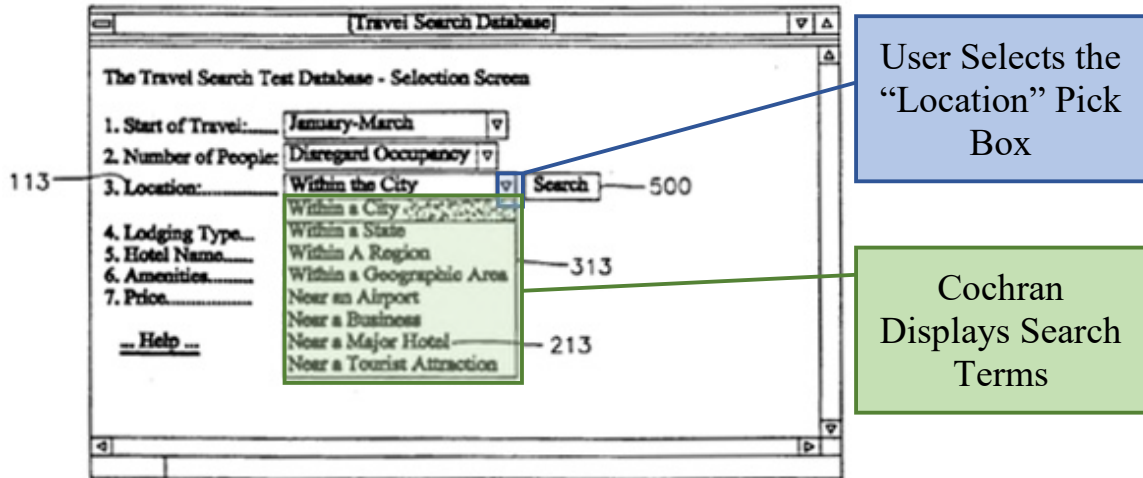
Cochran, Fig. 5b (annotated).

10. Claim 17

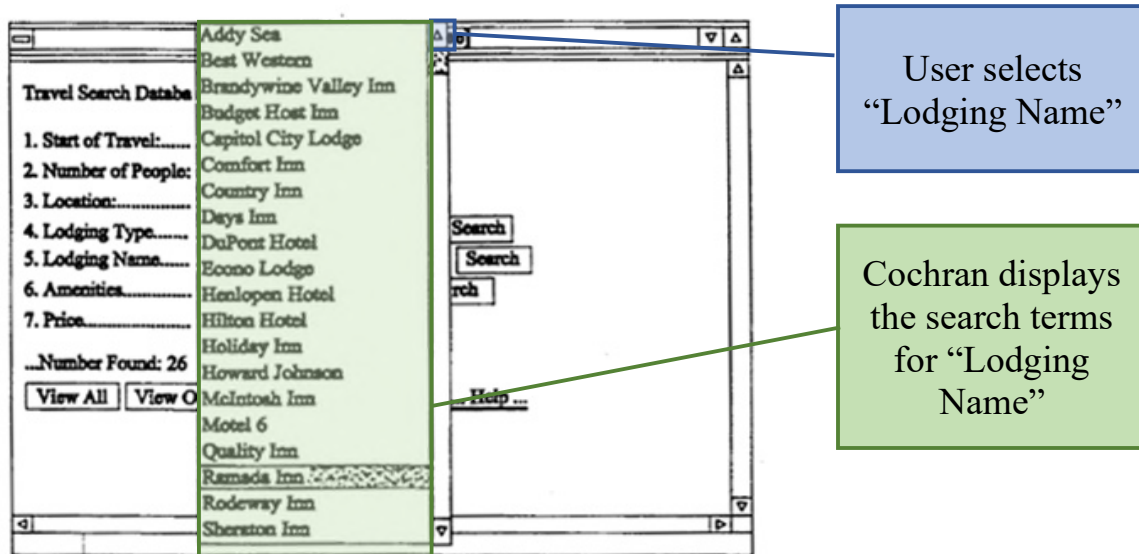
“The method [of] claim 3, further comprising displaying the list of data fields and a representation of the selected data field, wherein the representation of the selected data field comprises one or more elements from the selected data field.”

Cochran displays a set of list items and pick boxes (“*displaying the list of data fields and a representation of the selected data field*”), and also displays the search terms or elements underlying any selected pick box (“*wherein the representation of the selected data field comprises one or more elements from the selected data field*”). See §§XI.A.1.d, XI.A.1.e, XI.A.3.b. Specifically, Cochran teaches the use of “pick boxes” that use drop down menus to display search terms. Cochran, 7:36-47, 10:13-19, 10:43-11:65; Zadok, ¶¶270-274. A POSITA would have understood that the list of search terms is a “*representation of the selected data field*” comprising “*one or*

more elements from the selected data field” because it lists the elements from the given data field.



Cochran, Fig. 3d (annotated).

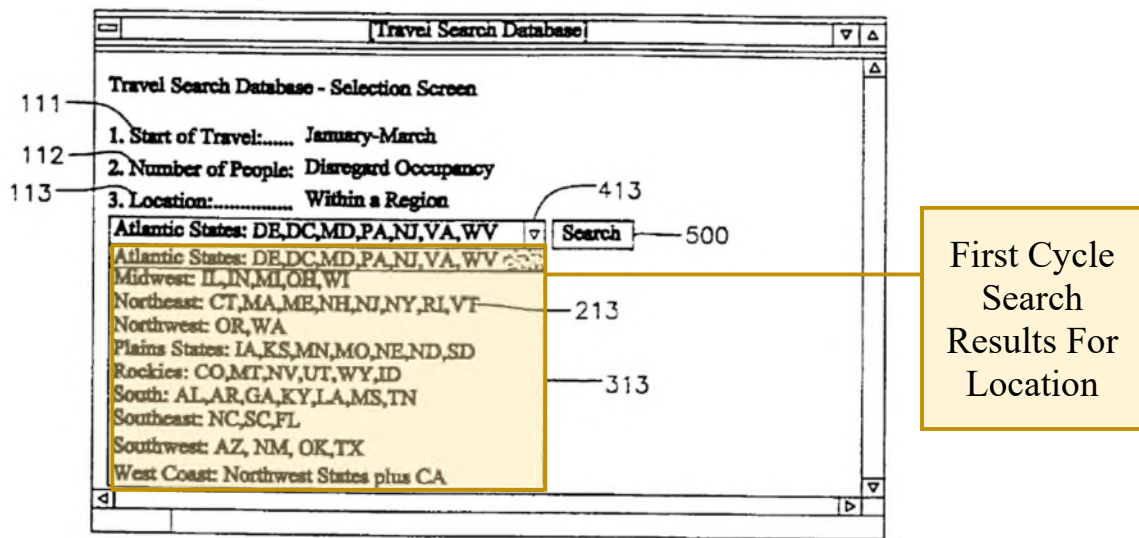


Id., Fig. 5d (annotated).

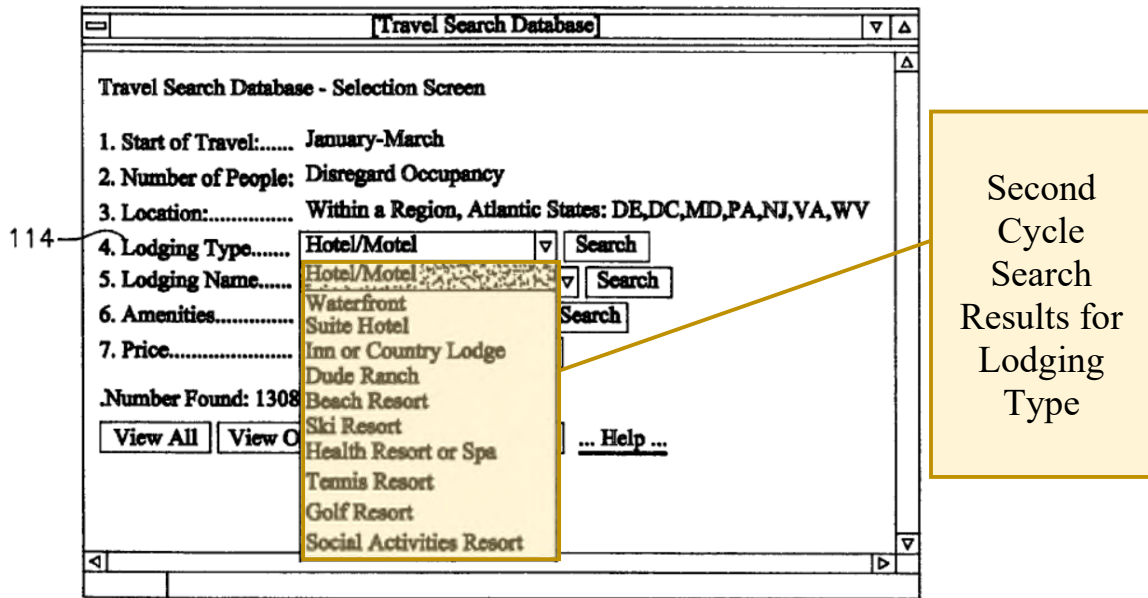
11. Claim 18

“The method of claim 3, wherein the method comprises executing a plurality of search cycles, each search cycle of the plurality of search cycles providing a search result list comprising selected data elements from the data base.”

Cochran allows a user to select from multiple pick boxes (“*executing a plurality of search cycles*”) wherein each pick box shows a list of search terms (“*each search cycle ... providing a search result list comprising selected data elements from the data base*”). Cochran, Abstract, 4:29-35, 8:45-64, 13:50-57, 14:64-66, Fig. 4d; Zadok, ¶¶275-286; §XI.A.9 (Claim 16). For example, the figures below show two search cycles:



Cochran, Fig. 4a (annotated), 12:27-43; *see also id.*, Figs. 3b, 3c, 10:66-11:16, 11:17-21.



Cochran, Fig. 4d (annotated), 13:50-57, 14:13-37.

The search terms displayed in the pick boxes are each cycle's "*search result list*." See, e.g., Cochran, Abstract, 4:20-50, 8:15-9:22, 12:66-13:13; 15:53-17:13. Specifically, "[t]he search terms contained in the lists are only those terms that are actually available in the database." *Id.*, Abstract. "Each search request will result in a subset of records" that "are then transmitted to ... the computer terminals" for display in the pick boxes. *Id.*, 6:40-52; Zadok, ¶¶276-286. Additional "search cycles" can be performed using the other pick boxes associated with "Lodging Type," "Lodging Name," "Amenities," and "Price." *Id.*, ¶283.

12. Claim 19

“The method of claim 18, further comprising resetting one or more search cycles of the plurality of search cycles.”

Cochran discloses a multi-cycle search process where a user selects search terms for data elements. §XI.A.11 (Claim 18); Zadok, ¶288. Cochran further discloses that the user can “reset all or part of the search” (“*resetting one or more search cycles of the plurality of search cycles*”) using the “Reset 4-7” or “Start Over” buttons. Cochran, Abstract, Fig. 5b; Zadok, ¶¶287-291.

When a user clicks “Reset 4-7,” Cochran “will clear any search terms previously selected for list identifiers 114-117 [(“Lodging Type,” “Lodging Name,” “Amenities,” and “Price”)] and replace them with default terms.” Cochran, 13:14-18. When a user presses “Start Over,” Cochran will “restart the selection process from the beginning.” *Id.*, 13:19-26. The user can reset “one or more” of the search cycles with either button. Zadok, ¶¶287-291.

13. Claim 20

a. [20.A]

“The method of claim 3 further comprising: determining a schema of the database;”

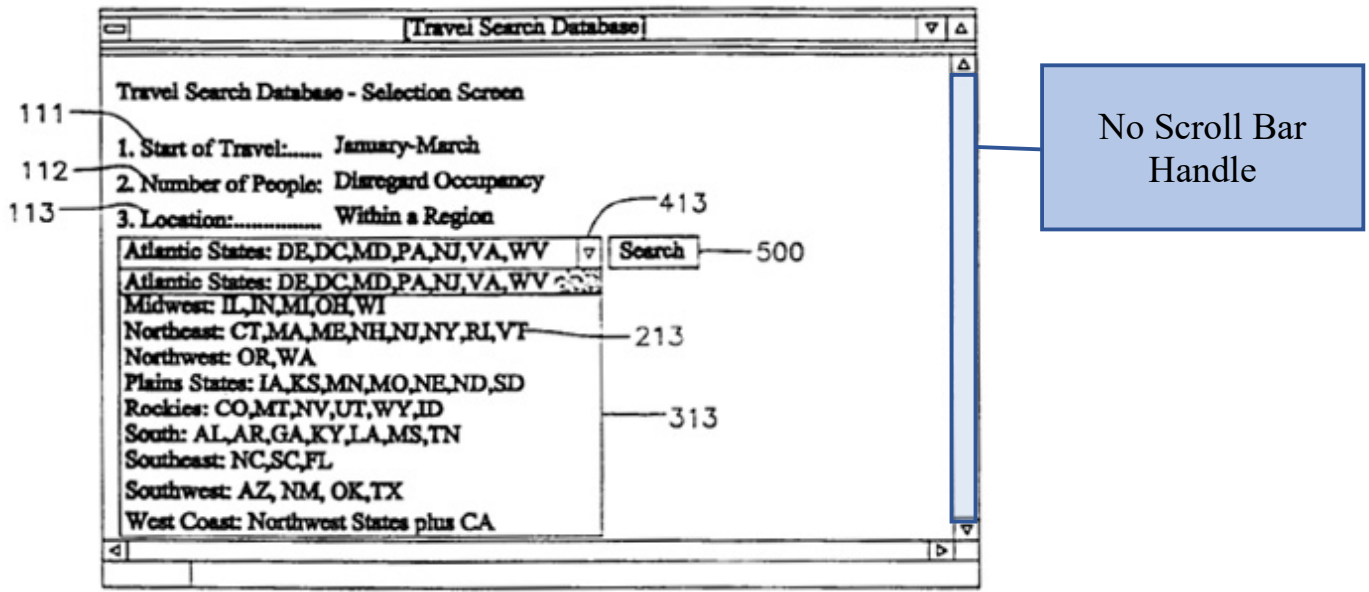
Cochran determines a schema of the database. *See* §XI.A.1.b; Zadok, ¶292.

b. [20.B]

“comparing the determined schema to a capability of a terminal for displaying the data; and”

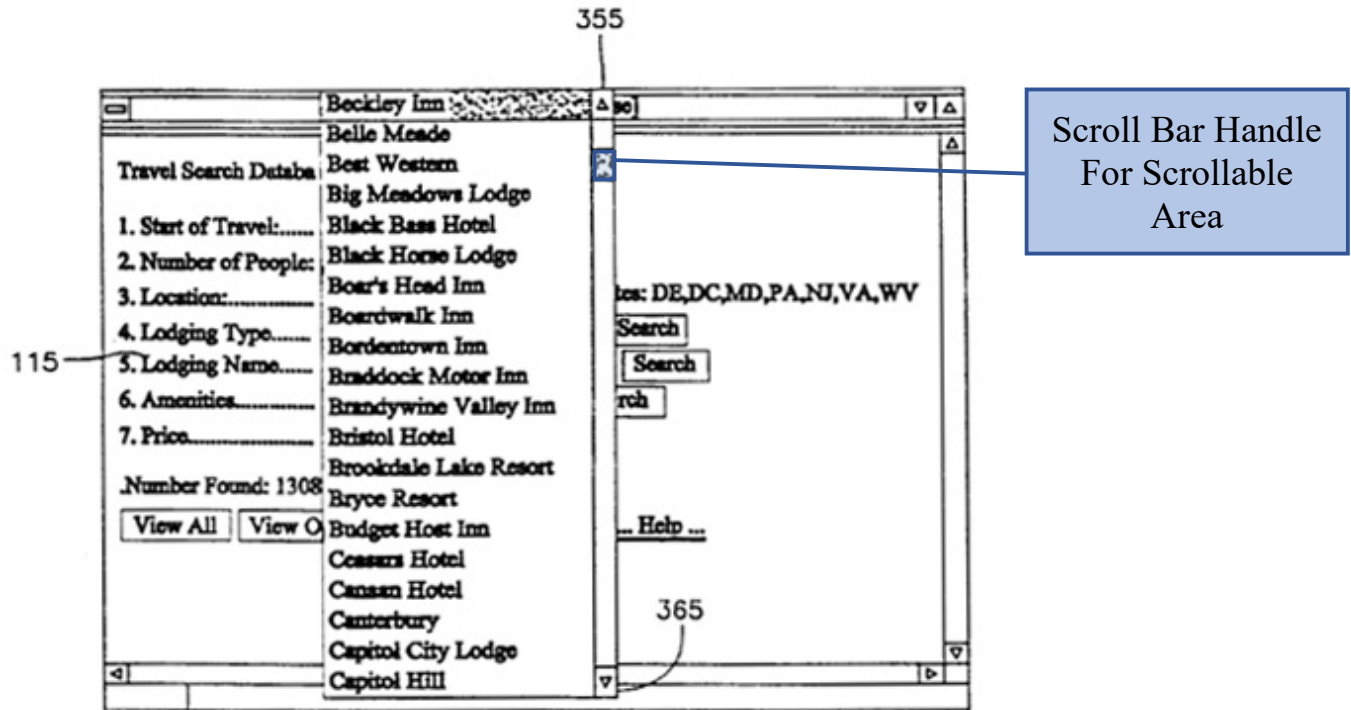
Cochran displays on screen a list of displayed labels and their accompanying pick boxes (“*the determined schema*”). See §§XI.A.1.c-XI.A.1.e, XI.A.3.b, XI.A.3.f; Zadok, ¶¶293-302. Because Cochran is meant to be used on “a computer terminal or other display device,” Cochran would obviously need a way to accommodate various display capabilities. Cochran, Abstract, 4:10-15, 6:11-12, Fig. 1. Cochran discloses various “computer terminal[s],” each with its own “display device” that can display the labels and pick boxes. *Id.*, 6:11-12, Fig. 1.

To accommodate various display devices, Cochran must decide whether to allow the user to scroll using the scroll bars on screen. For example, when a terminal’s display device can show everything simultaneously, Cochran displays a bar with no scroll bar handle:



Cochran, Fig. 4a (annotated).

A POSITA would have understood and found obvious, however, that when Cochran cannot display all labels and accompanying pick boxes on one page, Cochran would display a scroll bar including a handle. Zadok, ¶¶295-300. This is exactly what Cochran discloses for other elements such as pick boxes:



Cochran, Fig. 4e (annotated); *See also* §XI.A.1.e.

To accomplish this, Cochran obviously “*compare[s]*” how many labels (fields) and pick boxes must be displayed (“*the determined schema*”) to how much content can fit on the screen of each “computer terminal or other display device” (“*to a capability of a terminal for displaying the data*”). Zadok, ¶¶301-302.

c. [20.C]

“formatting a search request from the terminal to correspond to the determined schema.”

Cochran composes “search requests” (“*formatting a search request*”) using the search terms that the user selects via the pick boxes / search table (“*from the terminal*”). Cochran, 6:40-52, 7:5-8:65; Zadok, ¶¶303-306; *see also* §§XI.A.1.b-XI.A.1.d, XI.A.13. The search requests “correspond to the determined schema”; if

they did not, the search requests would not refer to the correct data fields in the database. *See* Cochran, 1:21-29; Zadok, ¶305; *see also* Cochran, 14:24-26 (“the server ... will search the subset of records for one additional term that is specified by one search button query”).

14. Claim 27

“The method of claim 3, further comprising displaying the data on a terminal, the terminal including one of a handheld device, a cellular phone, a geosynchronous positioning satellite (GPS) device, a wrist-worn device, an interactive phone device, a household appliance, a television, a television set top box, a handheld computer, a main frame computer and a personal computer.”

Cochran displays GUIs, including the list identifiers and search terms (“*displaying the data*”), on a user’s “computer terminal” that has a “display device” (“*a personal computer*”). *See, e.g.,* Cochran, Abstract, 4:10-19, 7:13-19, Zadok, ¶¶307-308.

15. Claim 28

“The method of claim 3, wherein the database is coupled to an Internet website, the method further comprising accessing the Internet website.”

Cochran’s system can be coupled to a server “accessed through a communication medium such as the Internet.” Cochran, 5:39-49. In one embodiment, the system is accessed through an Internet “Web Page addressed at ‘<http://travelsearch.com>.’” *Id.*, 6:31-64; *see also id.*, Fig. 1, 4:65-67, 6:6-39, 8:45-47. The server hosts the website and database. *Id.* A POSITA would have

understood that because Cochran may be accessed on an Internet website, and it stores its data in “database 712,” the database and Internet website are “*coupled*” together. Zadok, ¶¶309-311.

16. Claim 29

“The method of claim 3, wherein the database is coupled to a node in a telecommunications network, the method further comprising accessing the node in the telecommunications network.”

Cochran is implemented in software (a “webserver”) on a server that also hosts a database (“*the database is coupled to a node in a telecommunications network*”) and is accessed via the Internet (“*accessing the node in the telecommunications network*”). See §XI.A.15 (Claim 28); Zadok, ¶¶312-313. The webserver on Cochran’s “server” is a “node in a telecommunications network.” *Id.* That node is coupled to the database, which is also hosted on Cochran’s “server.” *Id.* Moreover, Cochran’s webserver is accessed via the Internet, which a POSITA would understand is a “*telecommunications network*.” See §XI.A.15 (Claim 28). Zadok, ¶¶312-313.

17. Claim 30

“The method of claim 3, wherein the database is coupled to a terminal for displaying the data.”

Cochran discloses accessing database information via a GUI on a web page displayed in a Web browser (“*the database is coupled to a terminal for displaying the data*”). Cochran, Abstract, 6:31-39. Cochran’s database is connected

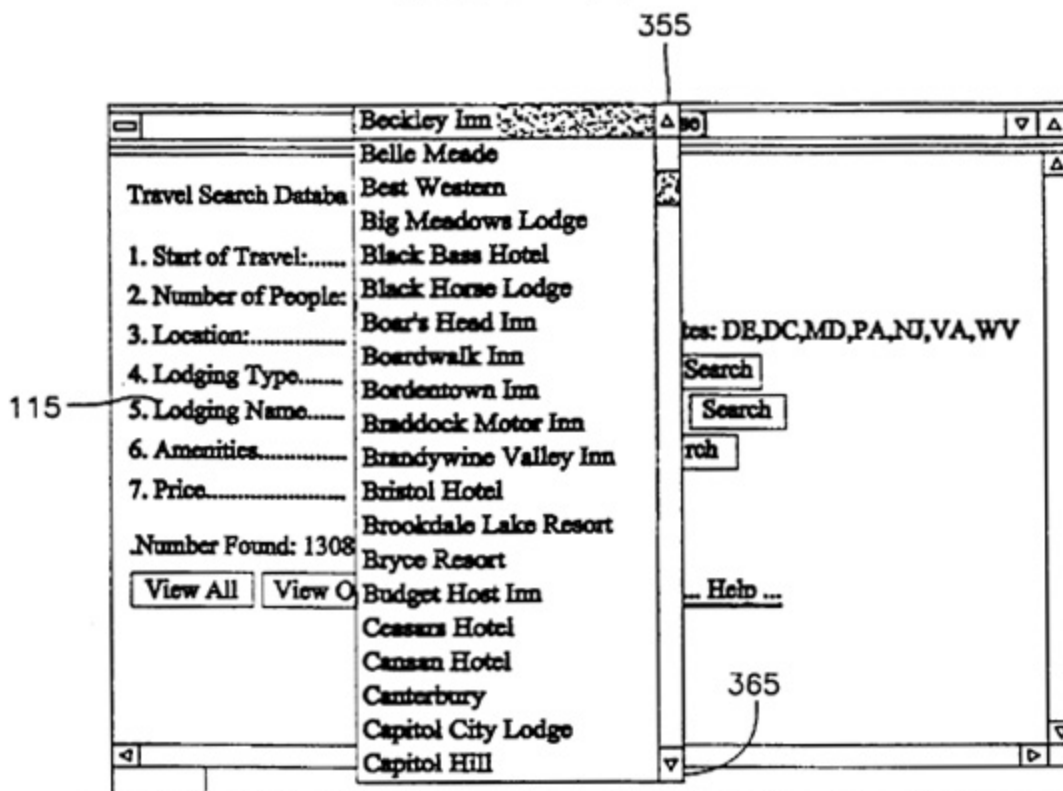
(“coupled”) to a computer terminal for displaying the data from the database (“terminal for displaying the data”). See, e.g., *id.*, 7:9-18. Cochran shows that the database data is displayed in a GUI on the computer terminal. *Id.*, Figs. 3a-10b. A POSITA would have understood that the database is coupled or connected to the computer terminal because “the server 710 transmits information ... to the user’s computer 720,” *id.*, 7:9-18, the server hosts the database, and the computer is the “terminal.” *Id.*, 6:31-39; Zadok, ¶¶314-317.

B. Motivation to Combine Cochran and Zellweger

A POSITA would have been motivated to combine Cochran and Zellweger. Zadok, ¶¶479-489.

Cochran endeavors to “provide an efficient and user-friendly method of searching computer databases” using “a search system that efficiently displays information to users.” Cochran, 4:4-19. Cochran acknowledges it may need to display more items than a pick box can display simultaneously, forcing a user to scroll through options. *Id.*, 10:58-65. A POSITA would have appreciated that certain lists in Cochran are extremely voluminous and would benefit from improved navigation methods:

FIG. 4e



Id., Fig. 4e (Lodging Names for 1308 records); Zadok, ¶¶482-485.

Zellweger sought to remedy this same problem. Zellweger, Abstract. Zellweger discloses “menus” well suited for use with lists with “hundreds, if not thousands of entries, [where] scrolling for a particular entry can be extremely tedious.” *Id.* A POSITA would have recognized that it is difficult for users to scroll through Cochran’s long pick box lists. Accordingly, a POSITA implementing Cochran’s system would have looked to Zellweger to improve the usability of the pick boxes. Zadok, ¶485. Specifically, a POSITA would have been motivated to create Zellweger Alphabet or Content Lists instead of scroll bars when, for example,

the number of available entries exceeds the number of entries that can be presented on a given screen. *Id.* Because Zellweger addresses this very problem, this combination would be nothing more than using one known method (Zellweger) to improve another known method (Cochran) according to the express teachings of Zellweger to yield predictable results. Moreover, Cochran expressly teaches that other methods for implementing pick boxes known in the art could be used to display “wide ranges” of terms. *See* Cochran, 10:61-65 (“The expansion of pick boxes to enable display of a wide range of terms may be done in accordance with routine programming steps ...”).

Moreover, Cochran and Zellweger are analogous art, directed to the same field (accessing information in databases), and address similar concepts (making information more easily accessible using graphical interfaces). They also both teach several of the same concepts (*e.g.*, GUIs to access databases) and are both implemented as computer software. *See, e.g.*, Cochran, Abstract, 1:14-20, 4:2-57; Zellweger, Abstract, 1:35-40, 2:30-43; Zadok, ¶¶480-481. A POSITA would have appreciated that Cochran’s and Zellweger’s teachings were well-known and complementary to one another. *Id.*

A POSITA would have had a high expectation of success in adapting Cochran in view of Zellweger’s teachings of the Alphabet and Content List menus. Zadok, ¶¶486-488. Cochran is exactly the type of system that Zellweger is intended

to improve. Both Cochran and Zellweger involve displaying database information in a GUI, and thus a POSITA would have understood that altering Cochran to truncate entries like in Zellweger's List menu would require nothing more than routine programming. *Id.*

XII. GROUND 2: HYPERSQL IN VIEW OF ZELLWEGER RENDERS OBVIOUS CLAIMS 1-5, 7, 15-19, 27-30⁴

A. Claim-by-Claim Analysis

1. Claim 1

a. [1.PRE]

“A method implemented on a computer for searching databases, comprising:”

HyperSQL discloses a system running on a Web server for creating “forms-and hypertext-based database interfaces for non computer experts” that “permit the user to query databases by filling out query forms selected from menus.” HyperSQL, 329-30, 335-36, Figs. 8, 9; Zadok, ¶¶321-323.

b. [1.A]

“determining a database schema for a database, wherein the database includes database fields;”

HyperSQL leverages tools used “for extracting the database schema from the database and automatically creating a working template form definition file” (“*determining a database schema for a database*”) such that “[n]o knowledge of

⁴ Zadok, ¶¶318-427.

SQL is required because the interface automatically generates SQL queries from the user's input." HyperSQL, 337, 330; Zadok, ¶324. A POSITA would have understood that the underlying databases include "database fields." *Id.*, ¶325; *see also* §XII.A.1.c (HyperSQL displaying said "database fields," including genus, species, etc. in a database of bacterium).

HyperSQL interfaces include query forms with a "form body." The form body "specifies what prompts and data entry fields" (*i.e.*, "*database fields*" in the "*database*") "should be presented to the user." HyperSQL, 333; Zadok, ¶326. The form body may include a variety of types of data entry fields, including "scrollable lists that are filled with information acquired automatically from the database" called "querylists." HyperSQL, 333; Zadok, ¶¶326-329.

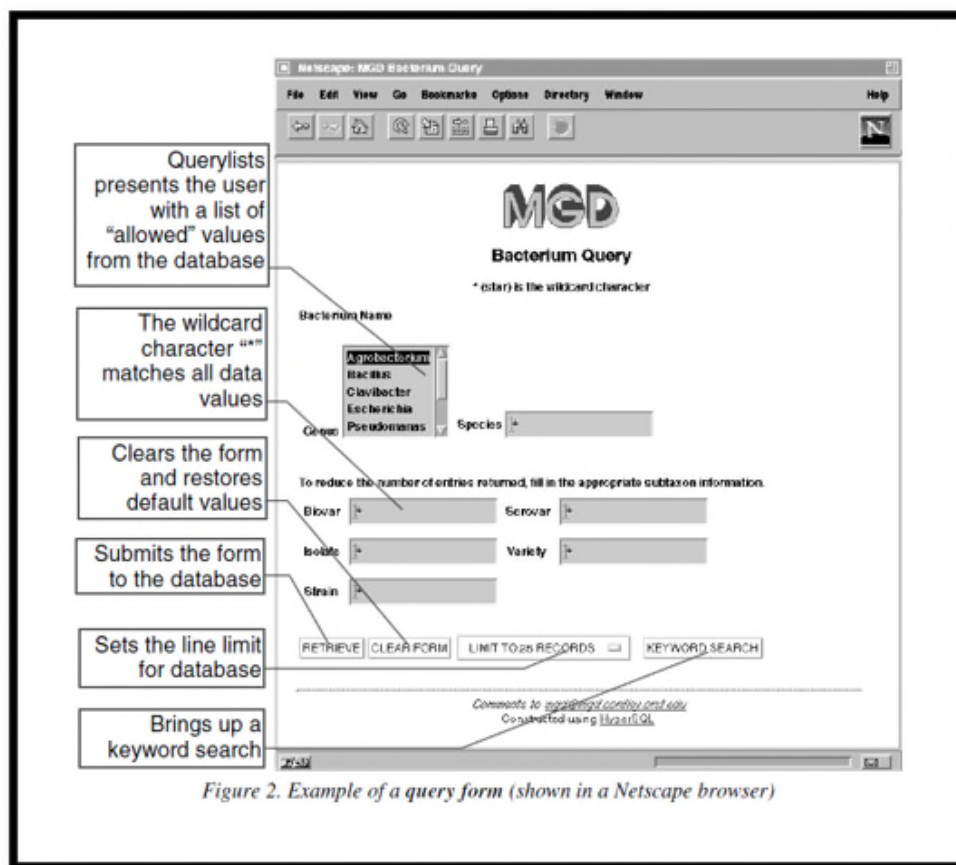


Figure 2. Example of a query form (shown in a Netscape browser)

HyperSQL, 331, Fig. 2. By using querylists, the user may “fill in fields” (*i.e.*, database fields) “incrementally by selecting from [the] querylists,” which each contain a “list of ‘allowed’ values from the database.” *Id.*, 331.

c. [1.B]

“providing a list of the database fields, wherein the list includes a descriptor indicating a data category;”

HyperSQL provides a list of “data entry fields” (“*database fields*”) with “prompts” (“*descriptors indicating a data category*”). Zadok, ¶¶330-332. HyperSQL discloses that the “form body section specifies what prompts [“*descriptors*”] and data entry fields [“*database fields*”] should be presented to the

user.” HyperSQL, 333. For example, in Figure 2, the displayed Querylist includes a “Genus” descriptor indicating that it relates to the database’s “Genus” data field:

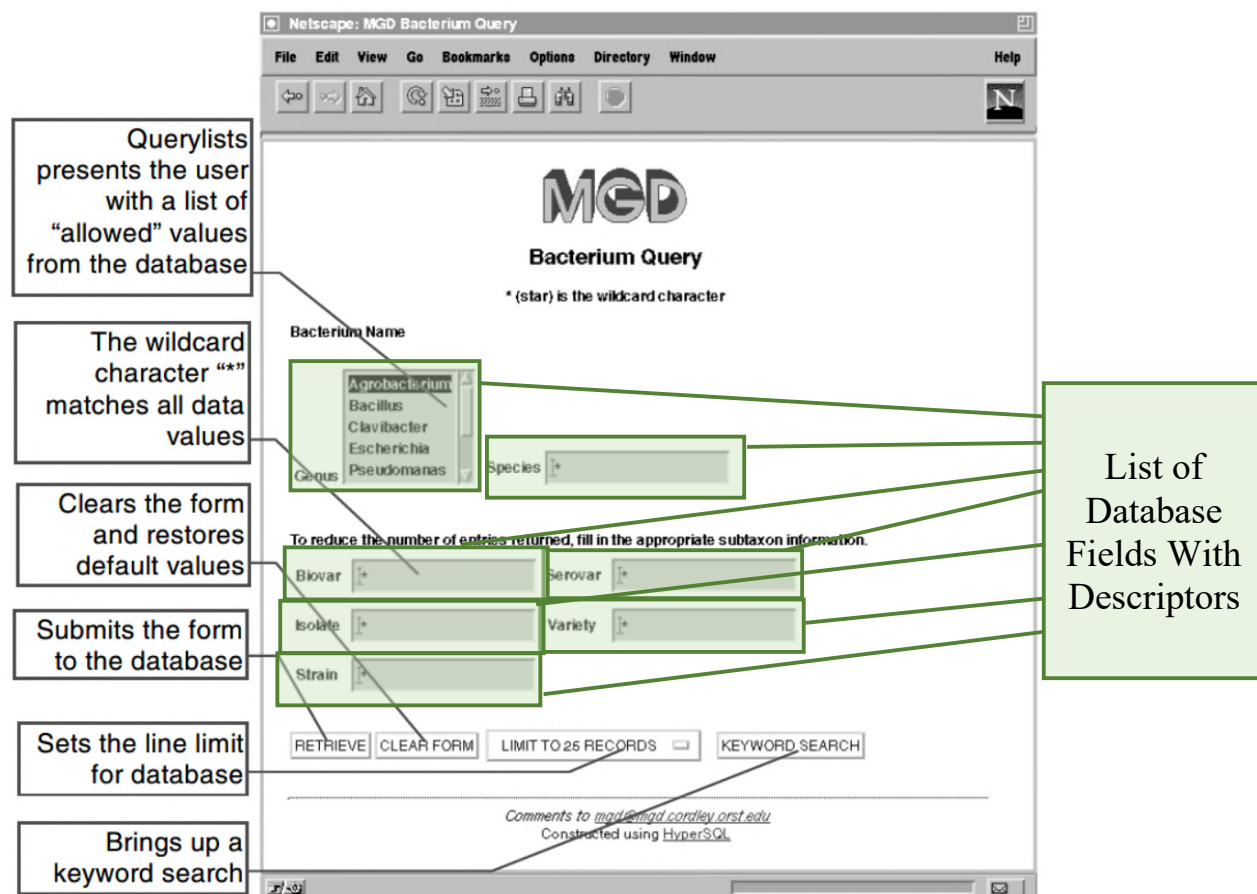


Figure 2. Example of a *query form* (shown in a Netscape browser)

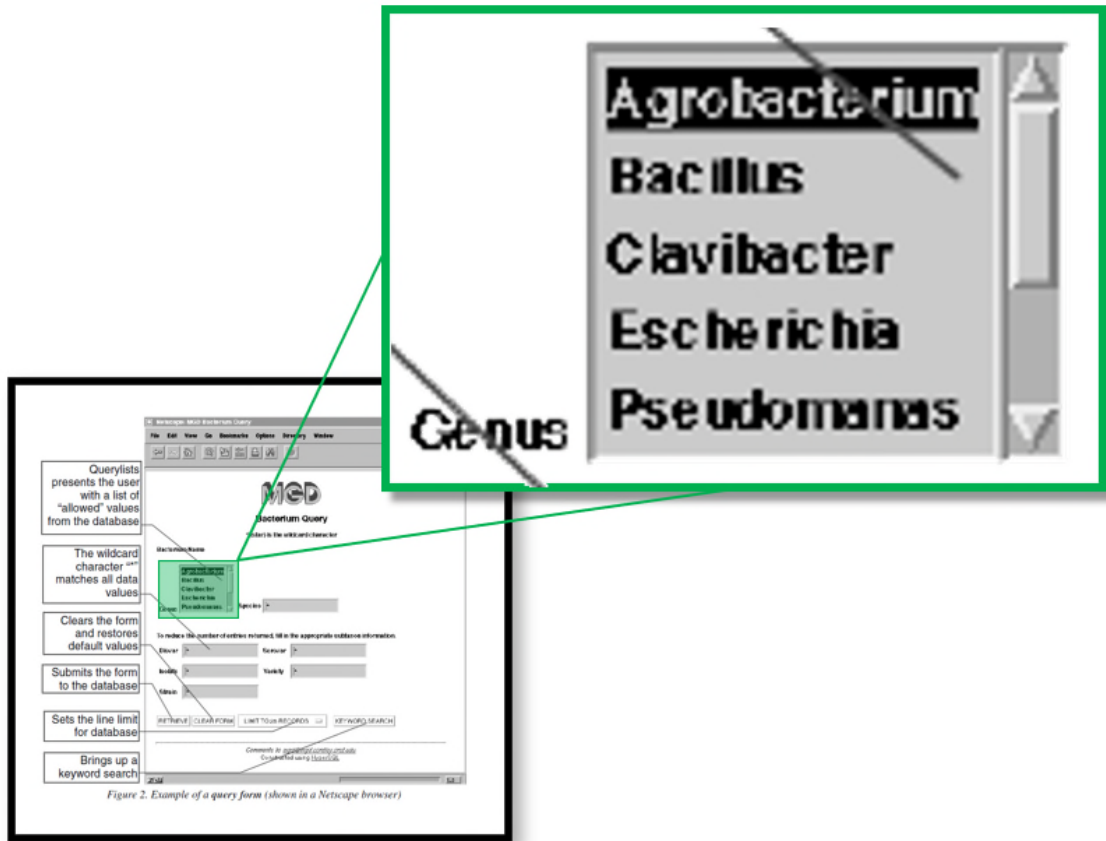
Id., Fig. 2 (annotated); see also *id.*, Fig. 5, 229, 331-32; §XII.A.1.b.

d. [1.C]

“receiving a search selection for a database field on the provided list of the database fields;”

HyperSQL users can select a database field (“receiving a search selection for a database field on the provided list of database fields”) by interacting with the field

on the query form. Zadok, ¶333. HyperSQL querylists are “scrollable lists that are filled with information acquired automatically from the database.” HyperSQL, 333.



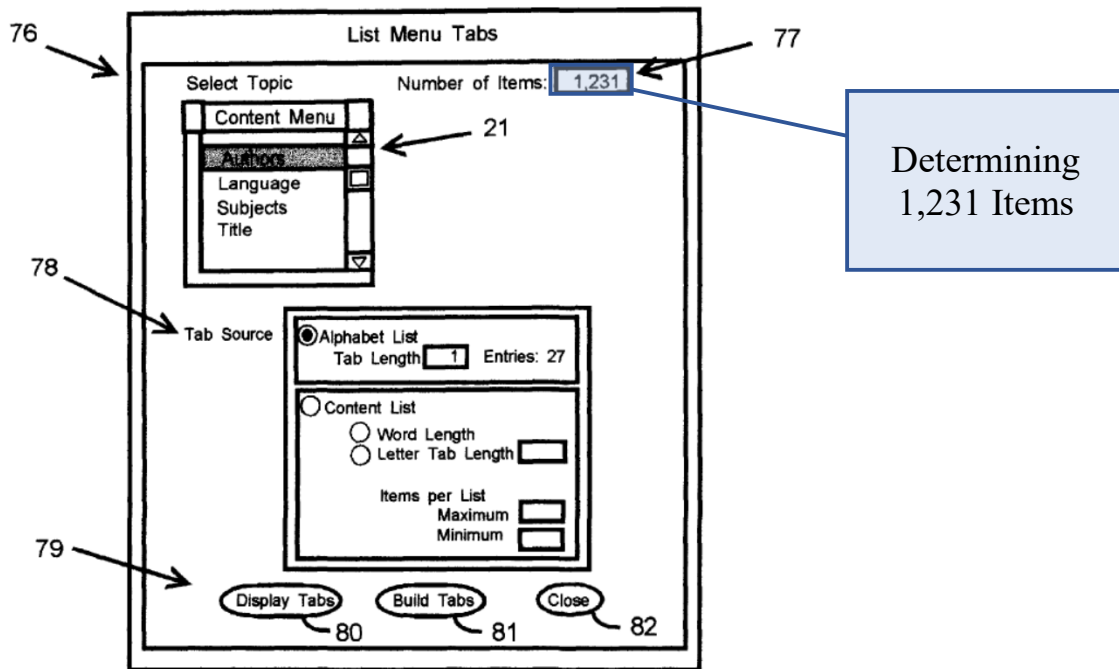
Id., Fig. 2 (annotated); *see also id.*, Fig. 5. Here, HyperSQL receives a search selection for database field “Genus” for which the “allowed” values from the database start with “Agrobacterium.” Zadok, ¶¶334-335.

e. [1.D]

“determining a quantity of entries in the selected database field;”

While HyperSQL does not mention that its system determines the quantity of items in a selected database field, a POSITA would have considered this obvious.

As explained in §XI.A.1.e, Zellweger discloses determining a number of items (“*quantity of entries*”) in a “*selected database field*.” Zadok, ¶¶336-343.



Zellweger, Fig. 6a (annotated). For instance, Zellweger’s system determines the number of entries in a database’s “Author” data field. *See id.* After applying this to HyperSQL, HyperSQL would similarly determine the number of entries populating a selected querylist (“*a quantity of entries in the selected database field*”). Zadok, ¶¶337-338.

f. [1.E]

“if the quantity exceed a specified amount, truncating data, and displaying the truncated data wherein the truncating reduces characters in one or more entries in the selected database field and the truncated data represents each of the entries in the selected database field; and”

HyperSQL does not expressly disclose “*truncating data*” if the quantity of entries for a selected field exceeds a specified amount. HyperSQL does teach employing the database schema to generate a query form with various input elements, including querylists. HyperSQL, 337, 330. A POSITA would have understood that—depending on the database—these querylists have the potential to become voluminous and unwieldy. Zellweger’s system is designed to handle such lists. Specifically, Zellweger discloses reducing (“*truncating*”) lists that are voluminous (*i.e.*, where “*the quantity [of entries] exceed[s] a specified amount*”) into Alphabet Lists or Content Lists. See §XI.A.1.f; Zadok, ¶¶344-348. As discussed above, the Alphabet and Content Lists represent all available entries using letters or tabs (“*the truncating reduces characters in one or more entries in the selected database field and the truncated data represents each of the entries in the selected database field*”). See §XI.A.1.f; Zadok, ¶¶348-354. HyperSQL in view of Zellweger further discloses that the user may set Zellweger’s “Minimum” setting to a “*specified amount*” such that if the number of entries to be displayed in the querylist is greater than the specified amount, the entries are truncated and displayed

in a Zellweger-style list using the truncated representation. *See* §XI.A.1.f; Zadok, ¶¶355-357. Thus, for example, if the number of entries in the “Genus” Querylist was large, the entries would be truncated to make the list more manageable. *See id.* Moreover, a POSITA would have understood that a user could select any threshold limit for converting lists to Alphabet or Content Lists. *Id.*

g. [1.F]

“if the quantity does not exceed the specified amount, displaying contents of the database field.”

As noted above, HyperSQL in view of Zellweger discloses or renders obvious truncating the list of entries if the list exceeds a specified number of entries. §XII.A.1.f. Accordingly, a POSITA would have understood or found obvious that the combined system displays the list without truncation (“*displaying contents of the database field*”) when the list does not exceed the specified amount (“*if the quantity does not exceed the specified amount*”). *Id.*; Zadok, ¶¶358-359.

2. Claim 2

“The method of claim 1, further comprising providing a key word search.”

HyperSQL’s query form includes a “KEYWORD SEARCH” button that, upon clicking, “brings up a form that accepts a text string to be located by searching all data fields” (“*providing a key word search*”). HyperSQL, 331, Fig. 2; Zadok, ¶¶360-362.

3. Claim 3⁵

a. [3.PRE]

“A method implemented on a computer for searching a database, comprising:”

HyperSQL in view of Zellweger discloses this limitation. *See* §XII.A.1.a (Claim [1.PRE]); Zadok, ¶363.

b. [3.A]

“generating a list of data fields;”

HyperSQL in view of Zellweger discloses this limitation. *See* §XII.A.1.c (Claim [1.B]); Zadok, ¶364.

c. [3.B]

“receiving a first data field selection from the list of data fields;”

HyperSQL in view of Zellweger discloses this limitation. *See* §XII.A.1.d (Claim [1.C]); Zadok, ¶366.

⁵ The discussions in §§XI.A.3.c-XI.A.3.e regarding the (a) equivalence of terminology in Claims 1 and 3 and (b) errors in Claim 3 also apply here. HyperSQL discloses these limitations for the same reasons. Zadok, ¶¶365, 367, 369.

d. [3.C]

“determining a first quantity indicative of a number of entries of the selected data field;”

HyperSQL in view of Zellweger discloses this limitation. *See* §XII.A.1.e (Claim [1.D]); Zadok, ¶368.

e. [3.D]

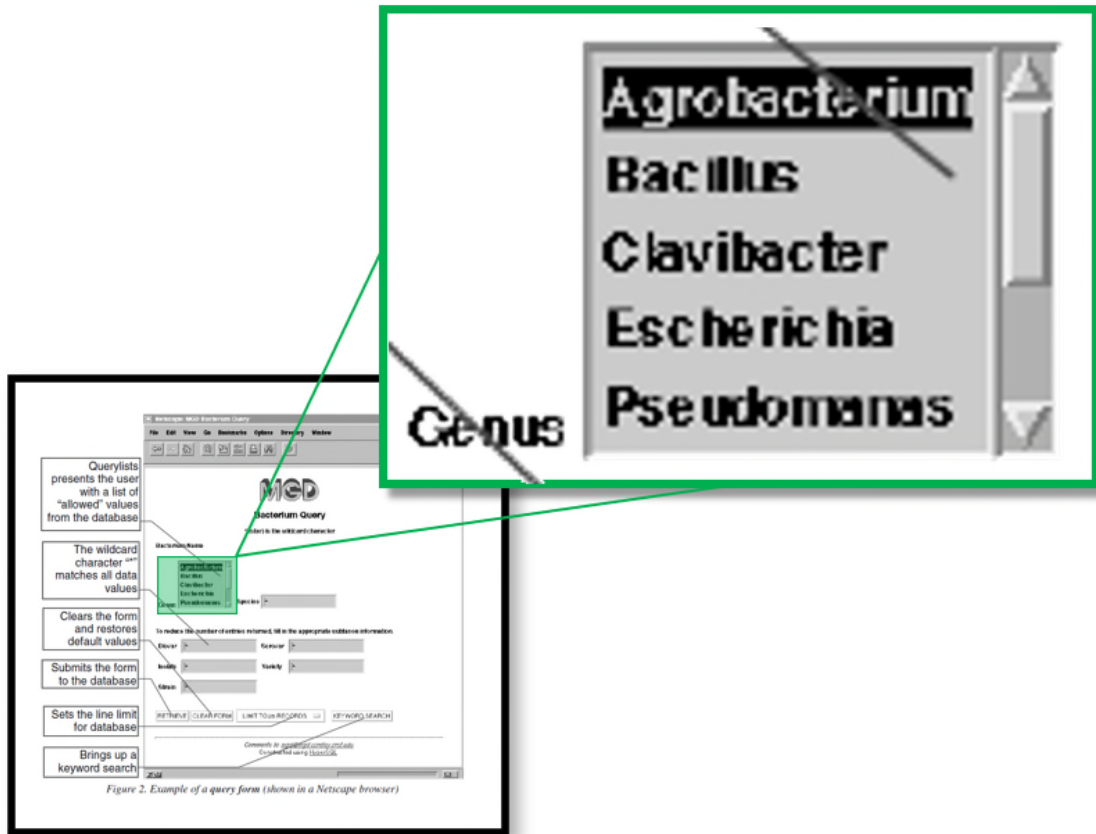
“if the first quantity exceeds a specified limit, reducing a size of data to be displayed from the selected data field wherein the reducing reduces characters in one or more entries in the selected data field and the size-reduced data represents each of the entries in the selected field; and”

HyperSQL in view of Zellweger discloses this limitation. *See* §§XII.A.1.f (Claim [1.E]), XII.A.3.f (Claim [3.E]); Zadok, ¶370.

f. [3.E]

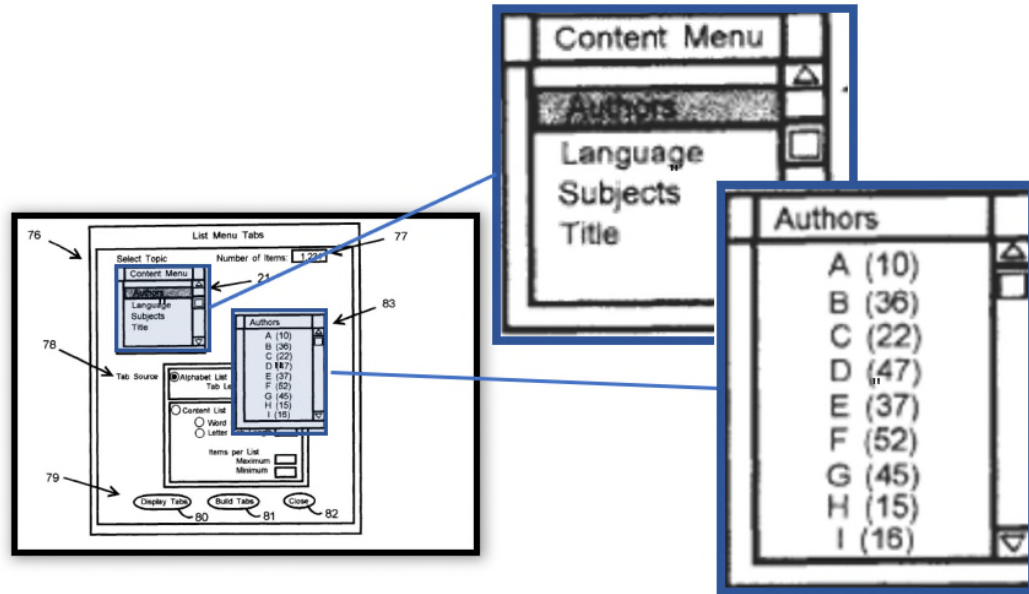
“displaying data from the selected data field.”

HyperSQL and Zellweger each display data in the querylist selected by the user (“*displaying data from the selected data field*”). *See* §XII.A.1.f (Claim [1.E]); Zadok, ¶¶371-377. HyperSQL displays the data in various inputs including querylists:

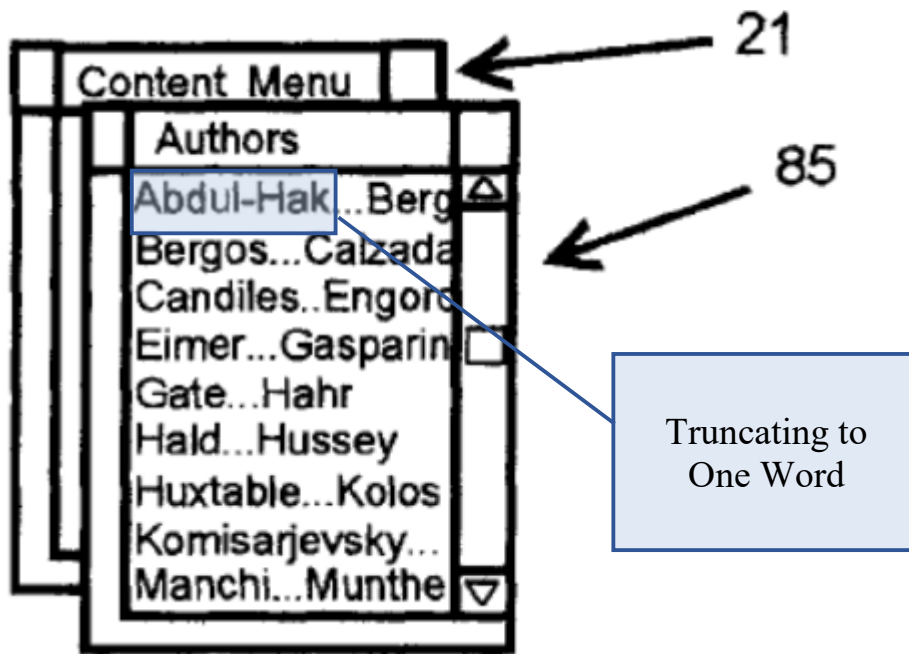


HyperSQL, Fig. 2 (annotated) (displaying data for the selected database field “Genus” in a querylist).

Zellweger displays entries in Alphabet List and Content List menus (“displaying data from the selected data field”):



Zellweger, Fig. 6b (annotated) (displaying data in an Alphabet List menu).



Id., Fig. 7 (annotated) (displaying data in a Content List menu).

4. Claim 4

“The method of claim 3, wherein the specified limit is fixed.”

HyperSQL may need to display many items in a querylist, which would require scrolling through a potentially lengthy list. *See* §XII.A.1.f; Zadok, ¶378. §XI.A.4 explains how Zellweger’s discloses this limitation for such voluminous lists. Zadok, ¶¶378-381.

5. Claim 5

“The method of claim 3, wherein the specified limit is variable.”

HyperSQL in view of Zellweger discloses this limitation. *See* §XI.A.5. Zadok, ¶¶382-383. §XI.A.5 discloses how Zellweger satisfies this limitation.

6. Claim 7

“The method of claim 3, wherein the specified limit is a user-determined limit.”

A Zellweger user “can specify the maximum and minimum” value for the Content List “items per list” (“*the specified limit is a user-determined limit*”). Zellweger, 4:26-31; §§XI.A.4-XI.A.5. Because the user specifies these values, they are “user-determined.” Zadok, ¶384.

7. Claim 15

a. [15.A]

“The method of claim 3, further comprising: receiving a first constraint, wherein the first constrain[t] is related to a data element in a data field; and”

A HyperSQL user may select a value from the provided querylists (“*receiving a first constraint*”). The “allowed values” represent actual entries for the data field associated with the querylist and, accordingly, a POSITA would have understood that any such value (“*constraint*”) is “*related to a data element in a data field*” for any given querylist. HyperSQL, 329-33, 333-34, Fig. 2; Zadok, ¶¶385-387.

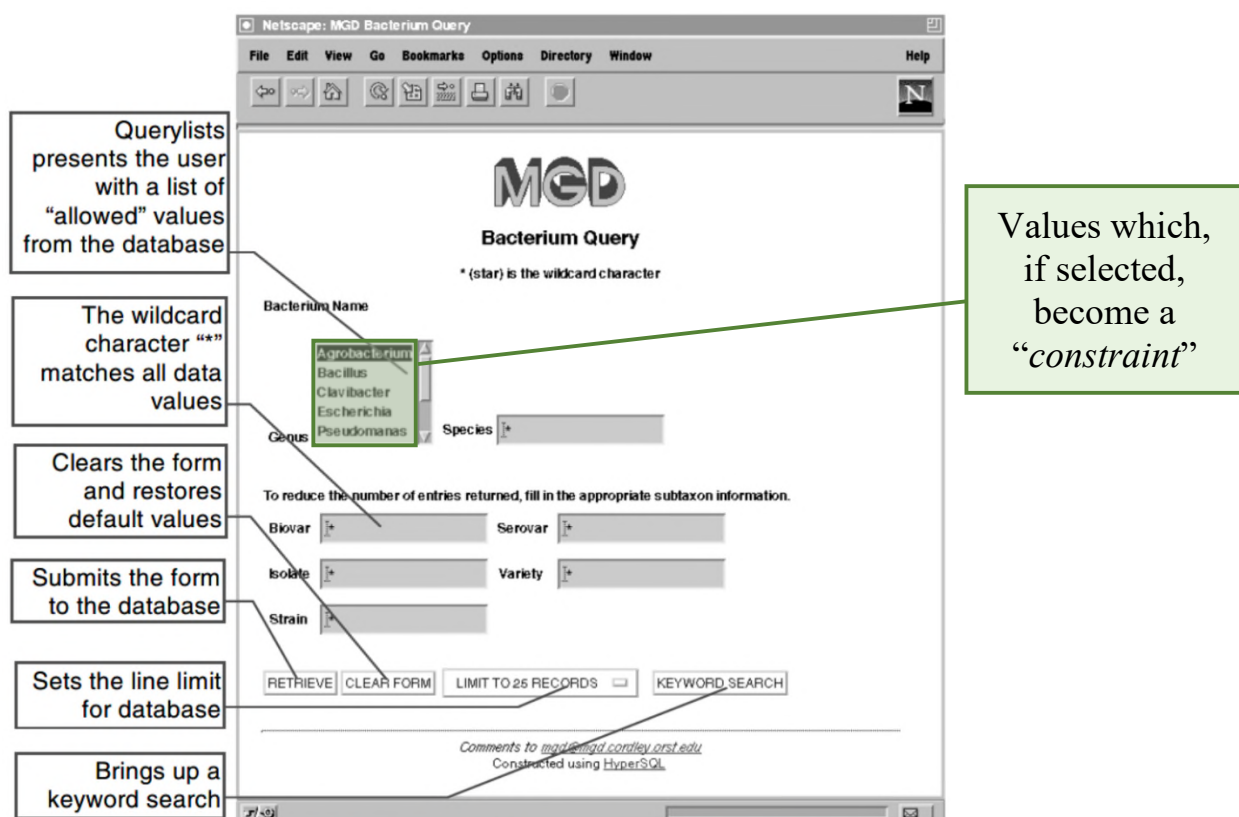


Figure 2. Example of a *query form* (shown in a Netscape browser)

HyperSQL, Fig. 2 (annotated). For example, a HyperSQL user may pick “*a first*

constraint” by clicking on “Agrobacterium” in the data field “Genus.” *Id.*, 332-33, Fig. 2; Zadok, ¶386.

b. [15.B]

“receiving one or more subsequent constraints, wherein search results are generated based on a combination of the first and the one or more subsequent constraints.”

HyperSQL teaches that a user may select a second value (“*subsequent constraints*”) and run a search that applies both the first and second values (“*wherein search results are generated based on a combination of the first and the one or more subsequent constraints.*”). See §XII.A.7.a (Claim [15.A]); Zadok, ¶¶388-393. For example, HyperSQL discloses a user picking the value “Agrobacterium” as the constraint for database field “Genus.” The user may then select subsequent constraints for other database fields like “Species,” “Biovar,” and “Serovar” (“*receiving one or more subsequent constraints*”) using text fields or additional querylists. HyperSQL, 332-33, Fig. 2.

When a user clicks on the “RETRIEVE” button on the query form, HyperSQL “[s]ubmits the form to the database” and the constraints are combined to produce results fitting the combined constraints (“*search results are generated based on a combination of the first and the one or more subsequent constraints*”). HyperSQL, 331-333, Figs. 2-4, 6; Zadok, ¶389.

Alternatively, multiple querylists may be used to allow for “query refinement” in which the user “fill[s] in fields incrementally by selecting from ‘querylists’” (“search results are generated based on a combination of the first and the one or more subsequent constraints”). HyperSQL, 331 (“Querylists are useful when the user does not know what the database contains, or simply wants to select from a list of values rather than typing.”); Zadok, ¶¶390-392.

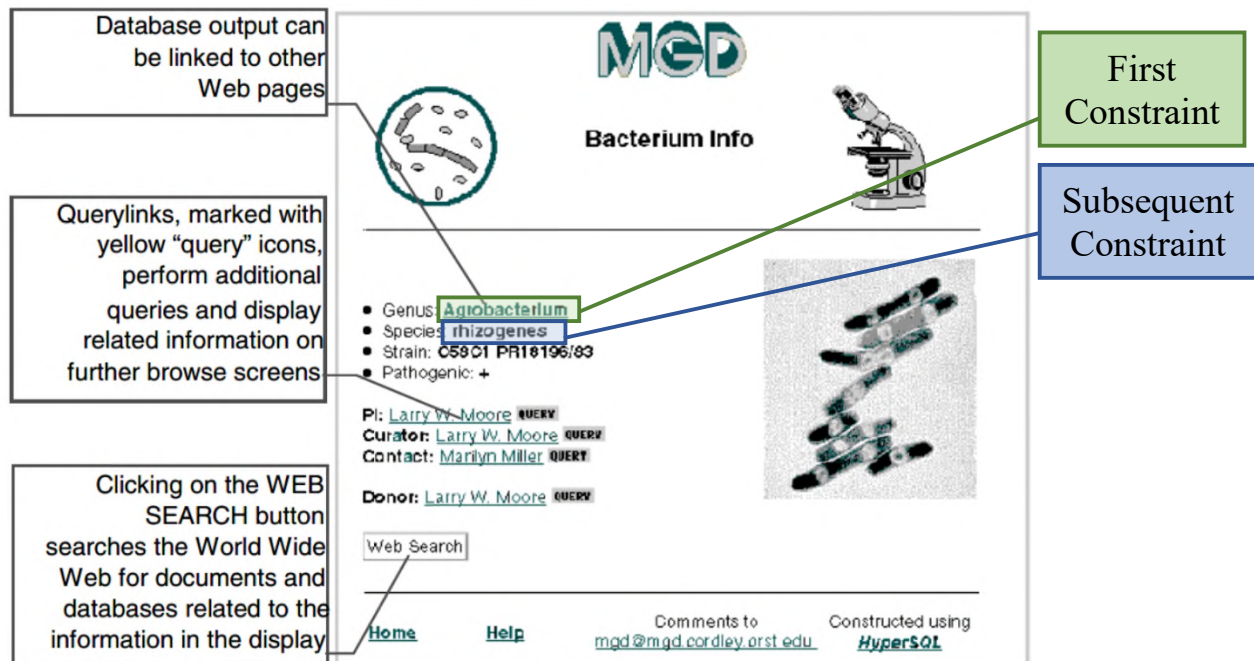


Figure 4. Example of a browse screen

HyperSQL, Fig. 4 (annotated); see also *id.*, Fig. 3.

8. Claim 16

“The method [of] claim 3, wherein each display of data comprises a search cycle, and further comprising receiving a refresh command, the refresh command resetting all search cycles.”

A HyperSQL user may perform a database search using values (entries/constraints) from multiple querylists (database fields). See §§XII.A.7.a, XII.A.7.b;

The screenshot shows the MGD Algal Query web interface. It features a banner with the MGD logo, a header with the title 'Algal Query', and a form body containing various input fields. The form includes a dropdown menu for 'Algae Name' with a list of genera (null, Acinetospora, Arthrocladia, Ascoseira, Bacheletia), a text field for 'Species', and three text fields for 'Isolate', 'Variety', and 'Strain'. Below these fields are three buttons: 'RETRIEVE', 'CLEAR FORM', and 'LIMIT TO 25 RECORDS' with a checkbox. The footer contains links for 'Home', 'Help', 'Comments to mqd@mqd.cordley.orst.edu', and 'Constructed using HyperSQL'. Labels on the right side of the image point to specific parts of the interface: 'BANNER' points to the MGD logo, 'HEADER' points to the 'Algal Query' title, 'FORM BODY' points to the 'Algae Name' dropdown, 'STANDARD BUTTONS' points to the 'RETRIEVE', 'CLEAR FORM', and 'LIMIT TO 25 RECORDS' buttons, and 'FOOTER' points to the footer links.

HyperSQL, Fig. 5 (showing multiple fields—each of which can be a querylist—for which a user may select values/entries/constraints). Each time a user selects one of the querylists/fields, HyperSQL searches the available values/entries and displays

them (“*wherein each display of data comprises a search cycle*”). See §XII.A.7 (Claim 15) (disclosing search using Genus and Species); HyperSQL, 331 (“query forms supports [sic] ‘query refinement,’ allowing user to fill in fields incrementally by selecting from ‘querylists’ (lists of allowed values).”), 333 (querylists are “filled with information acquired automatically from the database”); Zadok, ¶¶394-396.

HyperSQL further teaches a “CLEAR FORM” button which “erases any information entered into the form, replacing them with default values” (“*receiving a refresh command, the refresh command resetting all search cycles*”). HyperSQL, 330-31, Fig. 2; Zadok, ¶¶397-399.

9. Claim 17

“The method [of] claim 3, further comprising displaying the list of data fields and a representation of the selected data field, wherein the representation of the selected data field comprises one or more elements from the selected data field.”

HyperSQL teaches displaying multiple querylists (“*displaying the list of data fields*”) and allowing a user to select a particular querylist in order to display the allowed valued for that querylist (“*displaying . . . a representation of the selected data field [querylist], wherein the representation of the selected data field comprises one or more elements from the selected data field*”):

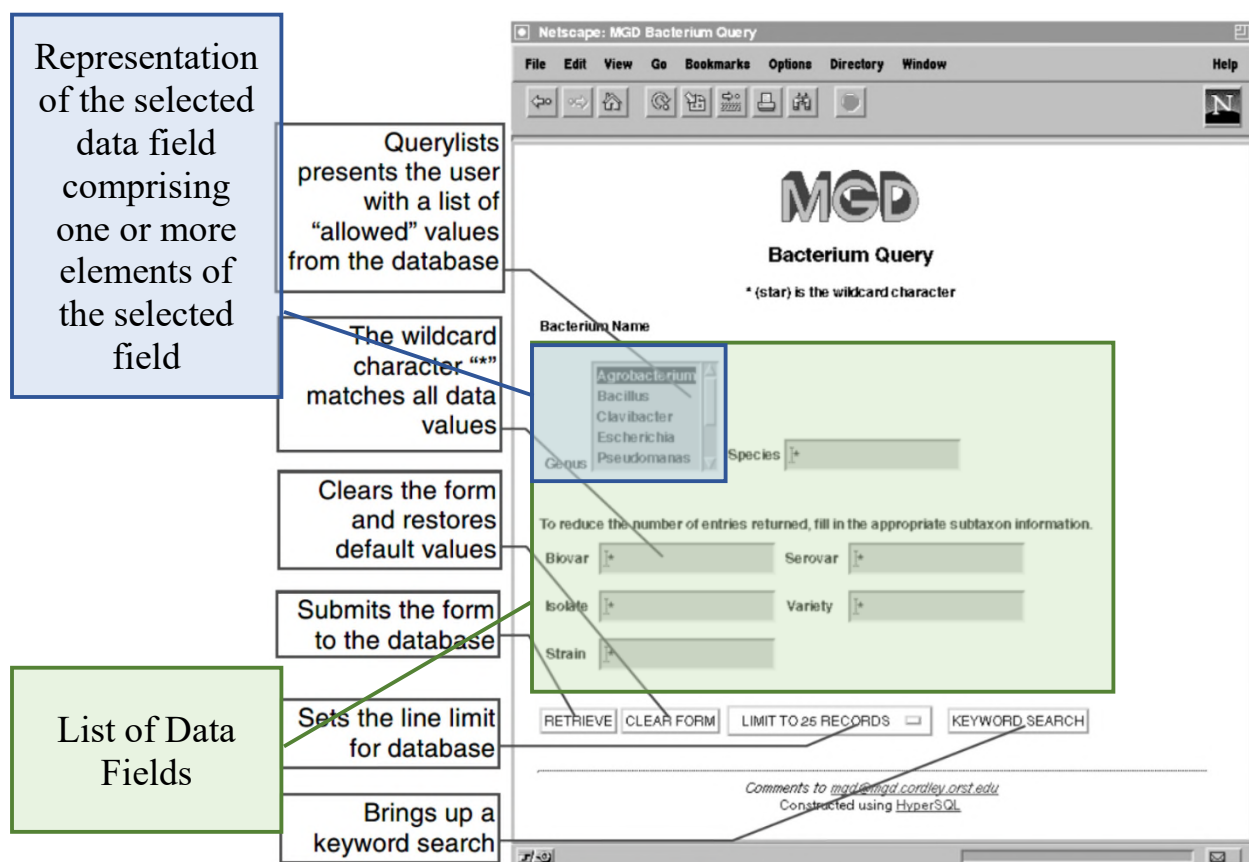


Figure 2. Example of a **query form** (shown in a Netscape browser)

HyperSQL, Fig. 2 (annotated); *id.*, 331-33, 338. A POSITA would have understood that HyperSQL’s list of allowed values is a “*representation of the selected data field*” comprising “*one or more elements from the selected data field*” because it lists the values for the corresponding database field. Zadok, ¶¶400-404.

10. Claim 18

“The method of claim 3, wherein the method comprises executing a plurality of search cycles, each search cycle of the plurality of search cycles providing a search result list comprising selected data elements from the data base.”

HyperSQL teaches selecting values (entries/constraints) from a plurality of querylists (database fields). Each time a user selects a different querylist (database

field) the system executes a “*search cycle*.” Specifically, HyperSQL teaches a “query refinement” feature. *See* §XII.A.8 (Claim 16); Zadok, ¶405.

When the user selects a value from a querylist, HyperSQL performs a search (“*search cycle*”) to populate other querylists with the “allowed values” “automatically from the database” (“*each search cycle of the plurality of search cycles providing a search result list comprising selected data elements from the data base*”). HyperSQL, 331, 333; Zadok, ¶406.

Accordingly, a POSITA would have understood that a user selecting values from multiple querylists as taught by HyperSQL is “*executing a plurality of search cycles*.” Zadok, ¶407.

11. Claim 19

“The method of claim 18, further comprising resetting one or more search cycles of the plurality of search cycles.”

HyperSQL discloses a multi-cycle search process where a user selects values from querylists. *See* §XII.A.10; Zadok, ¶409. HyperSQL further discloses that the query form has a “CLEAR FORM” button that resets the search cycles (“*resetting one or more search cycles of the plurality of search cycles*”). *See* §XII.A.8 (Claim 16); Zadok, ¶¶408-410.

12. Claim 27

“The method of claim 3, further comprising displaying the data on a terminal, the terminal including one of a handheld device, a cellular phone, a geosynchronous positioning satellite (GPS) device, a wrist-worn device, an interactive phone device, a household appliance, a television, a television set top box, a handheld computer, a main frame computer and a personal computer.”

HyperSQL discloses “a set of screens” which are “displayed by the browser” of a computer to display the “allowed” values in the querylists on the query forms (“*displaying the data on a terminal*”). HyperSQL, 330-31, 334-36, Figs. 2-6, 8-9; Zadok, ¶411. For example, the Web server sends database search results to “the browser for display” on that computer. HyperSQL, Fig. 9. A POSITA would have understood that a computer running a browser would be at least one of a “*handheld computer*” or a “*personal computer*.” Zadok, ¶412.

13. Claim 28

“The method of claim 3, wherein the database is coupled to an Internet website, the method further comprising accessing the Internet website.”

HyperSQL is implemented on the Web (Internet). Zadok, ¶413. HyperSQL, 329-30. Figure 9 confirms that the database communicates with a web server, where the website is located at the web server (“*the database is coupled to an Internet website*”). Zadok, ¶414.

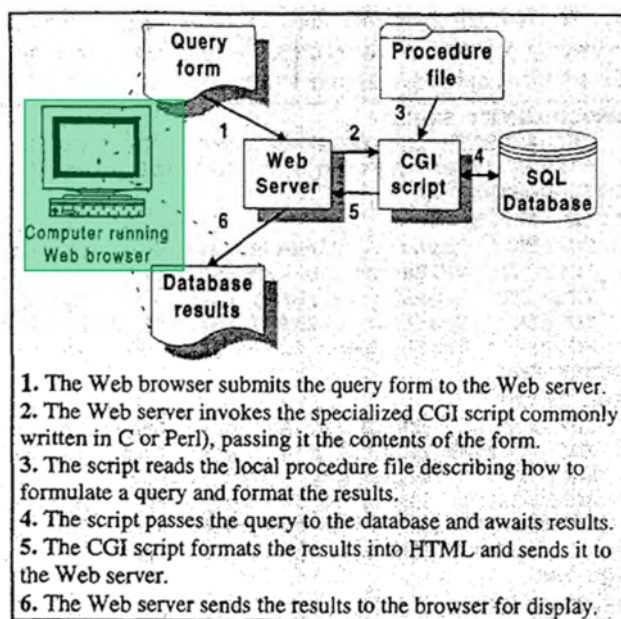


Figure 9. How CGI-based query interfaces work

HyperSQL, Fig. 9 (annotated); *see also id.*, Fig. 8, 333.

A HyperSQL user performs searches by accessing the website on a “Computer running Web browser” (“*accessing the Internet website*”). *Id.*, 329, Fig. 9; Zadok, ¶¶415-416.

14. Claim 29

“The method of claim 3, wherein the database is coupled to a node in a telecommunications network, the method further comprising accessing the node in the telecommunications network.”

The HyperSQL database(s) communicates with (“*is coupled to*”) other computers and devices (“*nodes*”), including the “HyperSQL gateway” (Fig. 8) and “Web Server” (Fig. 9) (“*the database is coupled to a node in a telecommunications network*”). *See* §XII.A.13. Users perform searches in the database by accessing those computers (“*nodes*”) over the Internet. (“*accessing the node in the*

telecommunications network”). See §XII.A.13; Zadok, ¶¶417-422. A POSITA would have understood that the Web (Internet) is a “*telecommunications network*.” *Id.*, ¶417.

15. Claim 30

“The method of claim 3, wherein the database is coupled to a terminal for displaying the data.”

HyperSQL is used to create “forms- and hypertext-based database interfaces for non computer experts.” HyperSQL, 329. The goal of HyperSQL is “to make it easier for scientists to retrieve information from [] databases using common Web browsers.” *Id.*, 330. Thus, the database communicates data to (“*the database is coupled to*”) a “Computer running Web browser” (“*a terminal for displaying the data*”). See *id.*, 329-30, 333, 335-36, Figs. 2-6, 9; Zadok, ¶¶423-427.

B. Motivation to Combine HyperSQL and Zellweger

A POSITA would have been motivated to combine HyperSQL and Zellweger. Zadok, ¶¶490-498.

HyperSQL’s goal is “to make it easier for scientists to retrieve information from remote scientific databases” by using a graphical “[q]uery interface.” HyperSQL, 329-30. HyperSQL uses “querylists” listing the allowed values from a database that a user may select as constraints in performing database searches. *Id.*, 330-31. HyperSQL’s exemplary use case is with biological databases where the querylists can be extremely lengthy and difficult to scroll through. *Id.*, 333, Fig. 2.

This is burdensome and makes selecting values in such querylists time consuming. Zadok, ¶¶493-494.

Zellweger sought to remedy this problem. Zellweger, Abstract. Zellweger discloses “menus” well suited for lists with “hundreds, if not thousands of entries, [where] scrolling for a particular entry can be extremely tedious.” Zellweger, Abstract. Accordingly, a POSITA would have looked to Zellweger to improve the usability of the HyperSQL querylists. Zadok, ¶495. A POSITA would have been motivated to create Zellweger Alphabet or Content Lists instead of scroll bars when, for example, the number of “allowed values” for a querylist exceeded a specified amount. *Id.* Because this is the exact problem addressed by Zellweger, this combination would be nothing more than using one known method (Zellweger) to improve a known system (HyperSQL) according to the express teachings of Zellweger to yield predictable results.

Moreover, HyperSQL and Zellweger are analogous art, directed to the same field (accessing information in databases), and address similar concepts (making information more easily accessible using graphical interfaces). They also both teach several of the same concepts (*e.g.*, GUIs to access databases) and are both implemented as computer software. *See, e.g.*, HyperSQL, 329-32, 335-36, Figs. 2-6, 8-9; Zellweger, Abstract, 1:35-40, 2:30-43; Zadok, ¶491. A POSITA would have

appreciated that HyperSQL's and Zellweger's teachings were well-known and complementary to one another. *Id.*, ¶492.

A POSITA would have had a high expectation of success in integrating Zellweger's list menu teachings into HyperSQL. Zadok, ¶¶496-498. Querylists are the exact types of menus that Zellweger is intended to improve. Because HyperSQL and Zellweger both involve displaying database information in a GUI, a POSITA would have understood that such an adaptation would require only routine programming. *Id.*

XIII. GROUND 3: HYPERSQL IN VIEW OF ZELLWEGER AND MICROSOFT ACCESS RENDERS OBVIOUS CLAIMS 21-23⁶

A. Claim-by-Claim Analysis

1. Claim 21

a. [21.A]

“The method of claim 3, wherein the method for searching the database comprises searching a first database and a second database, and further comprising:”

HyperSQL issues searches against two databases, a first “Sybase Database” and a second “Sybase Database” (“*the method for searching the database comprises searching a first database and a second database*”). HyperSQL, 335, Fig. 8; *see also id.*, 329-30; Zadok, ¶¶431-433.

⁶ Zadok, ¶¶428-451.

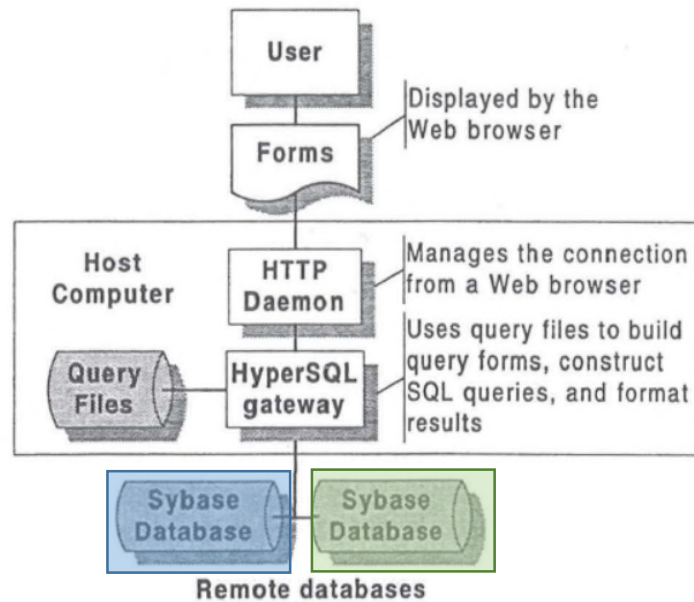


Figure 8. HyperSQL architecture

HyperSQL, Fig. 8 (annotated).

b. [21.B]

“receiving a first search result from the search of the first database;”

A POSITA would have understood that HyperSQL queries would have been sent to, and results returned from, the first Sybase Database (“*a first search result from the search of the first database*”). Zadok, ¶434; §XIII.A.1.a.

c. [21.C]

“receiving a second search result from the search of the second database; and”

HyperSQL in view of Zellweger and Microsoft Access disclose this limitation. See §§XIII.A.1.a, XIII.A.1.b. HyperSQL discloses that it supports “multi-database management capabl[e]” database systems. HyperSQL, 335.

HyperSQL discloses that it is used in situations “requiring data from multiple sources.” *Id.*, 329. A POSITA would have understood that in such a multi-database system, where data is stored in multiple sources, queries are sent to and fulfilled by different databases. Zadok, ¶¶435-436. A POSITA would have thus understood that HyperSQL queries are sent to, and results returned from, the second Sybase Database (“*a second search result from the search of the first database*”). *Id.*, ¶437; §XIII.A.1.a.

d. [21.D]

“combining the first and the second search results.”

A POSITA would have understood that HyperSQL discloses combining results from the first and second Sybase Databases (“*combining the first and the second search results*”). Zadok, ¶¶438-443. Because HyperSQL can use two databases, a POSITA would have understood that records responsive to a search may be stored in only one database (*e.g.*, when a particular collection of samples has unique samples in its database) or in both databases (*e.g.*, when multiple collections/donors have some of the same samples in their databases). *Id.*, ¶438; *See* §§XIII.A.1.b, XIII.A.1.c; *see also* HyperSQL, Fig. 3 (samples from three different collections/donors). The results from both databases would be combined together when a user presses the “RETRIEVE” button on the query form to “submit[] the form and return[] results.” *Id.*, 331.

Microsoft Access provides an example of how a POSITA would have implemented HyperSQL. Specifically, “[l]ogical operators (also called *Boolean operators*)” would be used “to combine the results of two or more comparison expressions” relating to HyperSQL’s different databases “into a single result” (“*combining the first and the second search results*”). Microsoft Access, 291.

2. Claim 22

“The method of claim 21, wherein the combining step comprises a Boolean AND operation.”

Microsoft Access discloses multiple “[b]oolean operators” including an “And” operation:

Table 9.3 Logical Operators			
Operator	Description	Example 1 Example 2	Result 1 Result 2
And	Logical and	True And True True And False	True False
Or	Inclusive or	True Or False False Or False	True False
Not	Logical not	Not True Not False	False True
Xor	Exclusive or	True Xor False True Xor True	True False

Microsoft Access, 291. The “And” operation would be implemented in the HyperSQL/Zellweger combination to create a result set containing results that exist in both Sybase Databases (“*the combining step comprises a Boolean AND operation*”), thus capturing *any redundant* results. Zadok, ¶¶444-445. This may occur, for example, when a user wants to identify samples that have been collected

by multiple donors (*e.g.*, where each donor/collection has its own database).

Id., ¶¶445-447; *see* HyperSQL, Figs. 3, 4.

3. Claim 23

“The method of claim 21, wherein the combining step comprises a Boolean OR operation.”

Microsoft Access discloses multiple “[*b*]oolean operators” including an “Or” operation:

Table 9.3 Logical Operators			
Operator	Description	Example 1 Example 2	Result 1 Result 2
And	Logical and	True And True True And False	True False
Or	Inclusive or	True Or False False Or False	True False
Not	Logical not	Not True Not False	False True
Xor	Exclusive or	True Xor False True Xor True	True False

Microsoft Access, 291. The “Or” operation would be implemented in the HyperSQL/Zellweger combination by returning results where each result is in either the first *or* the second Sybase Database (“the combining step comprises a Boolean OR operation”), thus capturing *every* result in either database. Zadok, ¶¶448-449. This may occur, for example, when a user wants to identify samples that have been collected by *any* donors (*e.g.*, where each donor/collection has its own database). *Id.*, ¶¶449-451; *see* HyperSQL, Figs. 3, 4.

B. Motivation to Combine HyperSQL, Zellweger, and Microsoft Access

A POSITA would have been motivated to combine HyperSQL and Zellweger, §XII.B, and to add Microsoft Access’s teachings to the combination. Zadok, ¶¶499-510.

HyperSQL discloses searching multiple databases. For example, biological samples may come from one or more collections or donors, and a POSITA would have understood that each collection may have its own database. *See* HyperSQL, Figs. 3, 4; Zadok, ¶501. A POSITA would have understood or found obvious that the results from these searches can be combined in various ways. *Id.*, ¶502. In some cases, for example, a user may want to identify samples that have been submitted by multiple donors/collectors for comparison. *Id.* A POSITA would have understood that this requires an “AND” operation. *Id.* Alternatively, the user may want to identify *any* samples that meet the proposed constraints. *Id.*, ¶503. A POSITA would have understood that this requires an “OR” operation. *Id.*

Moreover, Microsoft Access discloses ways in which database results could be combined using a well-known and commonly used database management system (DBMS), Microsoft Access. §§XIII.A.2, XIII.A.3; Zadok, ¶504.

Specifically, Microsoft Access discloses using Boolean operators like “AND” and “OR.” Microsoft Access, 291 (“*Logical operators* (also called *Boolean*

operators) are used most often to combine the results of two or more comparison expressions into a single result”);

Table 9.3 Logical Operators			
Operator	Description	Example 1 Example 2	Result 1 Result 2
And	Logical and	True And True True And False	True False
Or	Inclusive or	True Or False False Or False	True False
Not	Logical not	Not True Not False	False True
Xor	Exclusive or	True Xor False True Xor True	True False

Id. Accordingly, using these operations as taught by Microsoft Access to implement the “multi-database management” described by HyperSQL would be nothing more than implementing well-known and obvious techniques that are suggested by HyperSQL, known to a POSITA, and expressly taught by Microsoft Access. Zadok, ¶505.

Due to its popularity, it would have been obvious for a POSITA to implement the system of HyperSQL using teachings from Microsoft Access. Zadok, ¶¶506-507. Moreover, the processes taught for combining the results of searches from two or more databases were well-known to a POSITA regardless of the DBMS being used. *Id.*, ¶507. Finally, there are only so many ways to combine search results from two databases and using an “AND” or “OR” operation would be the most obvious. Accordingly, at a minimum this would be “obvious to try.” *Id.*, ¶508.

Further, HyperSQL, Zellweger, and Microsoft Access are analogous art, directed to the same field, and teach many of the same concepts. *See, e.g.*, §§X.B, X.C, X.D; Zadok, ¶500. A POSITA would have appreciated that HyperSQL's, Zellweger's, and Microsoft Access's teachings were well-known and complementary, making them good targets for combination. *Id.* For example, HyperSQL, Zellweger, and Microsoft Access relate to graphical means of accessing or displaying information in databases. *See, e.g.*, HyperSQL, 329; Zellweger, 2:1-18; Microsoft Access, 1-3.

A POSITA would have had a high expectation of success in adapting the HyperSQL/Zellweger combination to include Microsoft Access's teachings. Zadok, ¶509. A POSITA would have understood that the teachings being adopted from Microsoft Access comprise two of the most basic and fundamental digital logic operations: OR and AND. *Id.* A POSITA would have understood that implementing these teachings in the HyperSQL/Zellweger combination would have required only routine and minor software adjustments. *Id.*, ¶¶509-510.

XIV. GROUND 4: HYPERSQL IN VIEW OF ZELLWEGER, MICROSOFT ACCESS AND JANES RENDERS OBVIOUS CLAIM 25⁷

A. Claim-by-Claim Analysis

1. Claim 25

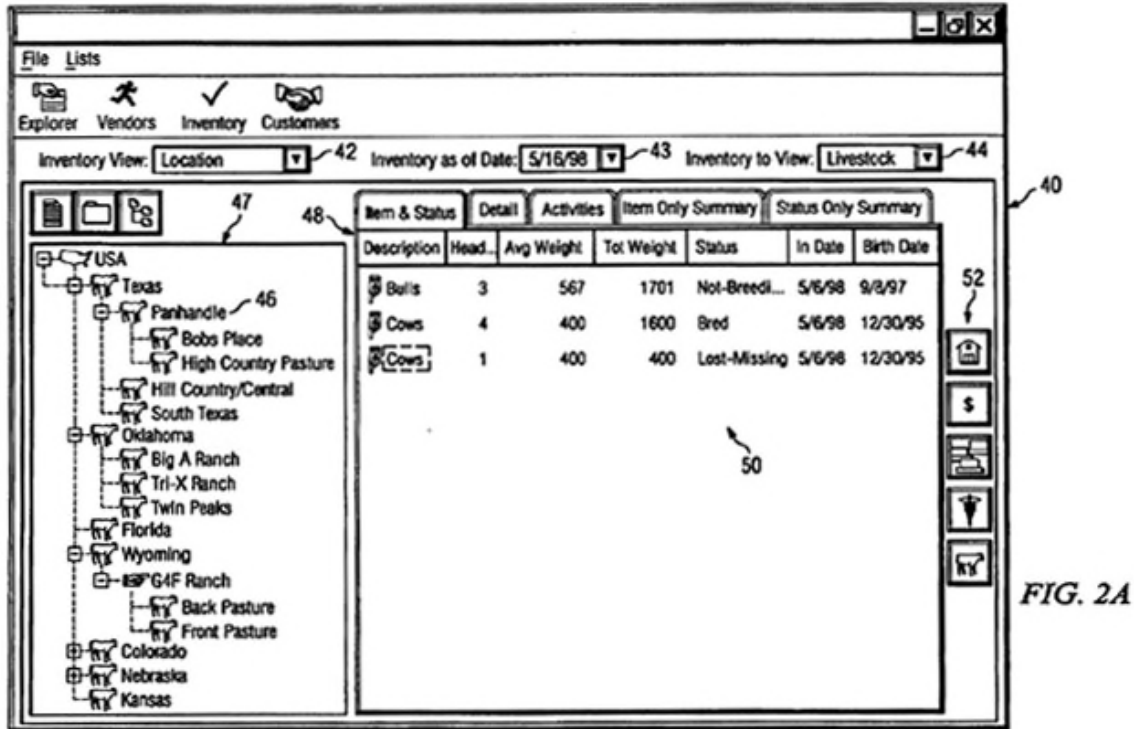
“The method of claim 21, wherein the first search result is represented by a first icon and the second search result is represented by a second icon, and wherein the combining step comprises overlaying the first and the second icons.”

HyperSQL’s “[d]atabase results are automatically formatted as graphics and hypertext,” meaning that the first result and second result (§§XIII.A.1.a–XIII.A.1.c) are each “formatted as graphics and hypertext” (*“the first search result is represented by a first icon and the second search result is represented by a second icon”*). Zadok, ¶455.

Janes discloses an “inventory system having a database storing inventory data” and “a graphical user interface operable to display data.” Janes, Abstract. The interface includes “displayed elements” such as “clickable icons and buttons.” Janes, 4:30-38, 8:61-9:11, 9:41-45, 11:44-55, 12:29-33. Users can associate elements in the graphical interface with other elements using a drag-and-drop system. *Id.*, 2:48-62. This “drag and drop method ... may be used to select any displayed element as the data source ... to any other displayed element as the data target” *Id.*, 12:34-44. Janes teaches an example implementation in the context

⁷ Zadok, ¶¶452-463.

of a livestock management system. *See e.g., id.*, Title, Abstract, 1:13-16, 2:10-19, Figs. 2A-2J, 4A-11.



Id., Fig. 2A (annotated). Farms and groups of farms are displayed as nodes in a hierarchical menu on the left and the inventory of the selected node is shown on the right. Each node and inventory item is displayed with an icon. *Id.*, 4:48-5:60.

A POSITA would have found it obvious to adapt the “drag and drop” style interface of Janes to the HyperSQL system. Such an interface would allow the user to search multiple databases/collections for results and combine those results using an easy-to-use graphical interface. Specifically, the combined system would allow for dragging and dropping an icon or text representation of one search result onto

the other search result to visually combine them (“*combining step comprises overlaying the first and the second icons*”). Zadok, ¶¶455-463.

B. Motivation to Combine HyperSQL, Zellweger, Microsoft Access, and Janes

A POSITA would have been motivated to combine HyperSQL, Zellweger, and Microsoft Access, §§XII.B, XIII.B, and to add Janes’ teachings to the combination. Zadok, ¶¶511-516.

As discussed above (§§XIII.A.1.a–XIII.A.3), the combined HyperSQL/Zellweger/Microsoft Access system teaches searching multiple databases and combining the results (*e.g.*, using AND/OR Boolean operations). The system, however, does not teach a specific GUI mechanism for requesting these operations. Zadok, ¶¶513-514. HyperSQL uses a graphical database interface intended to “make it easier ... to retrieve information from ... databases” and give more “meaningful feedback during the query process.” HyperSQL, 330. Accordingly, a POSITA implementing the system of HyperSQL would have turned to references like Janes (that teach GUIs for searching multiple databases) and been motivated to integrate Janes’ drag-and-drop teachings into the HyperSQL/Zellweger/Microsoft Access combination to allow a user to perform the combinations in HyperSQL’s GUI, giving the user more “meaningful feedback” and “mak[ing] it easier” (*i.e.*, more user-friendly) to conduct queries and combine the results. Zadok, ¶514.

Moreover, HyperSQL, Zellweger, Microsoft Access, and Janes are analogous art, directed to the same field, address the same general concept, and teach many of the same specific concepts. *See, e.g.*, §§X.B, X.C, X.D, X.E; Zadok, ¶512. A POSITA would have appreciated that HyperSQL's, Zellweger's, Microsoft Access's, and Janes' teachings were well-known and complementary, making them good targets for combination. *Id.* For example, HyperSQL, Zellweger, Microsoft Access, and Janes relate to graphical means of accessing or displaying information from databases. *See, e.g.*, HyperSQL, 329; Zellweger, 2:1-18; Microsoft Access, 1-3; Janes, Abstract, 2:27-48.

A POSITA would have had a high expectation of success in adapting the HyperSQL/Zellweger/Microsoft Access combination to include Janes' teachings. Zadok, ¶515. A POSITA would have understood that implementing Janes' teachings in the HyperSQL/Zellweger/Microsoft Access combination would have involved only routine software adjustments. *Id.*, ¶¶515-516.

XV. GROUND 5: COCHRAN IN VIEW OF ZELLWEGER, JANES, AND SPIEGEL RENDERS OBVIOUS CLAIM 26⁸

A. Claim-by-Claim Analysis

1. Claim 26

a. [26.A]

“The method of claim 3, wherein the displayed data includes one or more data records,”

Cochran displays data where the data includes “*data records.*” Zadok, ¶¶467-469. Cochran discloses a screen to “view the [search result] records” when the user clicks the “View All” button. Cochran, 13:2-14. The user may “scroll the screen” to view records if multiple records were returned pursuant to the user’s search. *Id.*

b. [26.B]

“each of the one or more data records represented by one or more of text title and an icon, and further comprising overlaying a text title or an icon on a symbol representing an online purchase,”

Janes discloses an inventory system where data records are “*represented by one or more of text title[s] and an icon.*” See §§X.E, XIV.A.1; Zadok, ¶470.

Spiegel discloses using dragging and dropping to make purchases online. Spiegel teaches a “computer method and system of electronic commerce.” Spiegel, Title, Abstract, 1:6-9. Spiegel recognizes that “[t]he World Wide Web portion of the Internet is especially conducive to conducting electronic commerce” and

⁸ Zadok, ¶¶464-477.

presents an improved “system for conducting electronic commerce” on the Internet. *Id.*, 2:1-20, 2:61-3:16. Spiegel suggests “dragging-and-dropping items into ... shopping carts” to provide a more user-friendly way to purchase items online. *Id.*, 10:31-54.

A POSITA would have augmented the Cochran/Zellweger combination with Spiegel’s teachings to allow users of Cochran’s Travel Search system to purchase stays at hotels. Cochran, 6:58-64. In this combination, Cochran’s records screen would display data records of the “hotels, resorts and inns” (which would obviously be designed to include icons and text as suggested by Janes). *Id.* A user would then drag-and-drop the icons and text onto a purchase icon (*e.g.*, the shopping cart in Spiegel) to purchase stays at one of the accommodations (“*overlaying a text title or an icon on a symbol representing an online purchase*”). Zadok, ¶¶471-474.

c. [26.C]

“wherein the step of overlaying initiates an online purchase of an item represented by the text title or icon.”

Spiegel discloses “dragging-and-dropping items into [] various shopping carts.” Spiegel, 10:43-54. The shopping cart is used to hold items for the “purchaser” to buy. *Id.*, 2:21-34. Spiegel discloses that a “shopping cart [may] already ha[ve] the appropriate billing and shipment information” associated with it. *Id.*, 4:19-24. Given that billing and shipping information is already associated with the shopping cart, a POSITA would have designed the system such that a purchase

is initiated upon a user dragging and dropping a product's text or icon onto the shopping cart (*"the step of overlaying initiated an online purchase of an item represented by the text title or icon"*) to make the process quicker and easier for users. Zadok, ¶¶475-477.

B. Motivation to Combine Cochran, Zellweger, Janes, and Spiegel

A POSITA would have been motivated to combine Cochran and Zellweger, §XI.B, and to add Janes' and Spiegel's teachings to the combination. Zadok, ¶¶517-522.

Cochran aims to "provide an efficient and user-friendly" way to search databases. Cochran, 4:2-19. A POSITA would have been motivated to find ways to further this goal. A POSITA would have recognized that integrating Janes' teachings of using icons and text for data entries (search results) would aid the user when "scroll[ing] the screen to view the records." Cochran, 13:2-7. For example, using icons would allow users to quickly discern different records—much like bullet points helps readers parse lists. Cochran, Fig. 4c; Zadok, ¶¶519.

A POSITA would have been further motivated to enable purchases in Cochran because Cochran's system is applied to commerce databases such as Cochran's exemplary Travel Search database. Cochran, 6:58-64. A POSITA would have understood that integrating Spiegel's teachings about dragging-and-dropping items onto shopping carts to assist in making purchases would increase Cochran's utility.

Zadok, ¶520. Specifically, allowing users to drag-and-drop icons or text for a search result onto another icon (*e.g.*, Spiegel’s shopping cart) to make a purchase (*e.g.*, booking a hotel in Cochran’s Travel Search embodiment) would further Cochran’s goal of providing the user with an easy-to-use solution. *Id.*

Further, Cochran, Zellweger, Janes, and Spiegel are analogous art, directed to the same field, and teach similar concepts. *See, e.g.*, §§X.A, X.C, X.E, X.F; Zadok, ¶518. A POSITA would have appreciated that Cochran’s, Zellweger’s, Janes’, and Spiegel’s teachings were well-known and complementary, making them good targets for combination. *Id.* For example, Cochran, Zellweger, Janes, and Spiegel relate to graphical means of accessing or manipulating information in databases. *See, e.g.*, Cochran, Abstract; Zellweger, 2:1-18; Janes, Abstract, 2:27-48; Spiegel, 6:59-7:13.

A POSITA would have had a high expectation of success in adapting the Cochran/Zellweger combination to include Janes’ and Spiegel’s teachings. Zadok, ¶521. A POSITA would have understood that implementing Janes’ and Spiegel’s teachings in the Cochran/Zellweger combination would involve only routine software adjustments to (1) include an icon for each of Cochran’s record on the view records screen and (2) add an additional purchasing icon that initiates a purchase after a user drags and drops a hotel’s text and/or icon onto the purchasing icon. *Id.*, ¶521-522.

XVI. SECONDARY CONSIDERATIONS OF NON-OBVIOUSNESS

There is no evidence of any secondary considerations of non-obviousness.
Zadok, ¶523.

XVII. CONCLUSION

Petitioner respectfully requests that review be instituted and that all
Challenged Claims of the 720 Patent be cancelled.

Dated: December 1, 2023

ORRICK, HERRINGTON & SUTCLIFFE LLP

By: /K. Patrick Herman/

K. Patrick Herman

Registration No. 75,018

Lead Counsel for Petitioner

Orrick, Herrington & Sutcliffe LLP

51 West 52nd Street

New York, NY 10019

T: (212) 506-3596

F: (212) 506-5151

Email: P52PTABDocket@orrick.com

*Attorney for Petitioner Oracle
Corporation*

CERTIFICATION PURSUANT TO 37 C.F.R §42.11

Pursuant to 37 C.F.R. §42.11, the undersigned hereby certifies that this Petition and accompanying evidence including Exhibits 1001 – 1007, 1009 – 1023 1025 – 1031 are not being presented for an improper purpose and that all legal contentions, allegations, and denials are warranted and have evidentiary support.

Dated: December 1, 2023

ORRICK, HERRINGTON & SUTCLIFFE LLP

By: /K. Patrick Herman/

K. Patrick Herman

Registration No. 75,018

Lead Counsel for Petitioner

Orrick, Herrington & Sutcliffe LLP

51 West 52nd Street

New York, NY 10019

T: (212) 506-3596

F: (212) 506-5151

Email: P52PTABDocket@orrick.com

CERTIFICATE OF COMPLIANCE

The undersigned certifies that the foregoing PETITION FOR *INTER PARTES* REVIEW complies with the type volume limitation in 37 C.F.R. §42.24(a)(1)(i). According to the utilized word-processing system's word count, the Petition—excluding the caption, table of contents, table of exhibits, mandatory notices, certification pursuant to 37 C.F.R. §42.11, certificate of word count, and certificate of service—contains 13,979 words.

Dated: December 1, 2023

ORRICK, HERRINGTON & SUTCLIFFE LLP

By: /K. Patrick Herman/
K. Patrick Herman
Registration No. 75,018
Lead Counsel for Petitioner

Orrick, Herrington & Sutcliffe LLP
51 West 52nd Street
New York, NY 10019
T: (212) 506-3596
F: (212) 506-5151
Email: P52PTABDocket@orrick.com

CERTIFICATE OF SERVICE

The undersigned hereby confirms that the foregoing Petition for *Inter Partes* Review and associated exhibits were caused to be served on December 1, 2023, via overnight courier upon the following counsel of record for Patent Owner:

John K. Harrop
P.O. Box 320171
Alexandria, VA 22320
harrop@vapatient.com

Copies of this Petition and accompanying exhibits were also served on December 1, 2023, via electronic mail on Patent Owner and on counsel of record for Vilox Technologies LLC in the related district court litigation involving DeBellis, Vilox Technologies, and Oracle Corporation:

Joseph L. DeBellis, M.D.
282 Edge of Woods Road
Southampton, NY 11968
josephdebellis@gmail.com

Richard A. Forsten
Saul Ewing LLP
1201 North Market St., Ste. 2300
Wilmington, DE 19801
richard.forsten@saul.com

/Karen Johnson/

Karen Johnson

APPENDIX A – CLAIM LISTING

Claim 1:

[1.PRE] “A method implemented on a computer for searching databases, comprising:”

[1.A] “determining a database schema for a database, wherein the database includes database fields;”

[1.B] “providing a list of the database fields, wherein the list includes a descriptor indicating a data category;”

[1.C] “receiving a search selection for a database field on the provided list of the database fields;”

[1.D] “determining a quantity of entries in the selected database field;”

[1.E] “if the quantity exceed a specified amount, truncating data, and displaying the truncated data wherein the truncating reduces characters in one or more entries in the selected database field and the truncated data represents each of the entries in the selected database field; and”

[1.F] “if the quantity does not exceed the specified amount, displaying contents of the database field.”

Claim 2:

“The method of claim 1, further comprising providing a key word search.”

Claim 3:

[3.PRE] “A method implemented on a computer for searching a database, comprising:”

[3.A] “generating a list of data fields;”

[3.B] “receiving a first data field selection from the list of data fields;”

[3.C] “determining a first quantity indicative of a number of entries of the selected data field;”

[3.D] “if the first quantity exceeds a specified limit, reducing a size of data to be displayed from the selected data field wherein the reducing reduces characters in one or more entries in the selected data field and the size-reduced data represents each of the entries in the selected field; and”

[3.E] “displaying data from the selected data field.”

Claim 4:

“The method of claim 3, wherein the specified limit is fixed.”

Claim 5:

“The method of claim 3, wherein the specified limit is variable.”

Claim 6:

“The method of claim 3, wherein the data are displayed on a terminal, and wherein the specified limit is determined dynamically, based on a characteristic of the terminal.”

Claim 7:

“The method of claim 3, wherein the specified limit is a user-determined limit.”

Claim 15:

[15.A] “The method of claim 3, further comprising: receiving a first constraint, wherein the first constraint is related to a data element in a data field; and”

[15.B] “receiving one or more subsequent constraints, wherein search results are generated based on a combination of the first and the one or more subsequent constraints.”

Claim 16:

“The method [of] claim 3, wherein each display of data comprises a search cycle, and further comprising receiving a refresh command, the refresh command resetting all search cycles.”

Claim 17:

“The method [of] claim 3, further comprising displaying the list of data fields and a representation of the selected data field, wherein the representation of the selected data field comprises one or more elements from the selected data field.”

Claim 18:

“The method of claim 3, wherein the method comprises executing a plurality of search cycles, each search cycle of the plurality of search cycles providing a search result list comprising selected data elements from the data base.”

Claim 19:

“The method of claim 18, further comprising resetting one or more search cycles of the plurality of search cycles.”

Claim 20:

[20.A] “The method of claim 3 further comprising: determining a schema of the database;”

[20.B] “comparing the determined schema to a capability of a terminal for displaying the data; and”

[20.C] “formatting a search request from the terminal to correspond to the determined schema.”

Claim 21:

[21.A] “The method of claim 3, wherein the method for searching the database comprises searching a first database and a second database, and further comprising:”

[21.B] “receiving a first search result from the search of the first database;”

[21.C] “receiving a second search result from the search of the second database; and”

[21.D] “combining the first and the second search results.”

Claim 22:

“The method of claim 21, wherein the combining step comprises a Boolean AND operation.”

Claim 23:

“The method of claim 21, wherein the combining step comprises a Boolean OR operation.”

Claim 25:

“The method of claim 21, wherein the first search result is represented by a first icon and the second search result is represented by a second icon, and wherein the combining step comprises overlaying the first and the second icons.”

Claim 26:

[26.A] “The method of claim 3, wherein the displayed data includes one or more data records,”

[26.B] “each of the one or more data records represented by one or more of text title and an icon, and further comprising overlaying a text title or an icon on a symbol representing an online purchase,”

[26.C] “wherein the step of overlaying initiates an online purchase of an item represented by the text title or icon.”

Claim 27:

“The method of claim 3, further comprising displaying the data on a terminal, the terminal including one of a handheld device, a cellular phone, a geosynchronous positioning satellite (GPS) device, a wrist-worn device, an interactive phone device, a household appliance, a television, a television set top box, a handheld computer, a main frame computer and a personal computer.”

Claim 28:

“The method of claim 3, wherein the database is coupled to an Internet website, the method further comprising accessing the Internet website.”

Claim 29:

“The method of claim 3, wherein the database is coupled to a node in a telecommunications network, the method further comprising accessing the node in the telecommunications network.”

Claim 30:

“The method of claim 3, wherein the database is coupled to a terminal for displaying the data.”