**PATROLL Winning Submission**

**U.S. Patent 6,760,720**

U.S. Patent 6,760,720 (“*Vilox Technologies*” or the “patent-at-issue”) was filed on February 25, 2000, and claims priority on the same date. Claim 1 of the patent-at-issue is generally directed to a method implemented on a computer for searching databases. This method involves several key steps: initially, it determines a database schema for a particular database, which encompasses database fields. A list of these fields is provided, including a descriptor that indicates the category of data each field contains. Upon receiving a search selection for one of these database fields, the method then ascertains the quantity of entries within the selected field. If this quantity exceeds a predetermined threshold, the method proceeds to truncate the data, effectively reducing the number of characters in one or more entries within the selected database field, yet ensuring the truncated data still represents each entry in that field. Conversely, if the number of entries does not surpass the specified limit, it displays the full contents of the database field as is.

The primary reference, U.S. Patent 6,189,004 (“*Rassen*”), was filed on May 6, 1998, and claims priority on the same date. The patent is directed to a method for automatically defining a query interface for a datamart which includes the steps of accessing a schema description and a query interface description. The datamart, as used in this patent, is a data warehouse or a database. The schema description outlines the organization of data in a database, detailing the relationships between the fact tables and dimension tables within the datamart, while the query interface description delineates the fields related to the schema description that can be utilized in queries, along with instructions on how results should be displayed to the user.

The secondary reference, U.S. Patent 6,370,518 (“*Google*”), was filed on October 5, 1998, and claims priority on the same date. The patent is directed to an improved query input technique that enhances the efficiency and speed with which users can access and display information from a structured database. This technique allows a user to input a query using just a few characters, enabling the retrieval and display of at least a portion of a record. With each character entered by the user, the system presents a progressively reduced list of indexes that match the entered characters. Once the user identifies the desired index among the displayed options, they can stop entering characters and instead use a zoom window to scroll through and explore the full record. To aid in usability, various visual feedback indicators are provided, offering intuitive guidance throughout the process. This method significantly reduces the number of keystrokes needed to find and display pertinent information, making database interactions both quicker and more user-friendly.

A sample claim chart comparingclaim 1 of *Vilox Technologies* to *Rassen* and *Google* is provided below.

|  |  |
| --- | --- |
| **US6760720** (“*Vilox Technologies*”) | **A.**  **US6189004** (*“Rassen”*)  **B.**  **US6370518** (*“Google”*) |
| 1.pre. A **method implemented on a computer for searching databases**, comprising: | **A.**  **US6189004**  “One embodiment of the invention includes a **method for automatically defining a query interface for a datamart**.” *Rassen* at col. 2:65-66  “**Datamart** or Data Warehouse—**is a database**.” *Rassen* at col. 5:35  **B.**  **US6370518**  “The present invention pertains to an improved query input technique in which a user supplies a relatively few character-based query entries and is able to **retrieve and display at least as portion of a record from a structured database**.” *Google* at col. 2:38-41  “1. A method for **displaying a record from a structured database** on a display screen of an electronic device having a reduced-size input interface, the structured database including a plurality of records, the records including at least one field and being indexed by indexes comprising n characters, the method comprising:” *Google* at Claim 1 |
| 1.a. **determining a database schema for a database, wherein the database includes database fields**;  *(cont.)*  1.a. **determining a database schema for a database, wherein the database includes database fields**; | **A.**  **US6189004**  “The method comprises **accessing a schema description and a query interface description for the datamart**. The schema description specifies a schema, which in turn, defines the relationships between the fact tables and dimension tables of the datamart. The **query interface description specifies the fields, related to the schema description**, that can be used in a query and the way in which results are to be presented to the user.” *Rassen* at col. 2:67 through col. 3:1-7  “1. A method of generating a datamart from a plurality of sources having a query mechanism interface using a computer system, the method comprising: **accessing a schema definition which describes a schema for the datamart**, the schema defined using a number of semantic meanings describing transformations of data between the plurality of sources and the datamart; **accessing a description of the query mechanism interface to be generated in the datamart**; . . . generating the query mechanism interface from the query mechanism description and the description of the schema.” *Rassen* at Claim 1  **B.**  **US6370518**  “1. A method for displaying a record from a structured database on a display screen of an electronic device having a reduced-size input interface, the **structured database including a plurality of records, the records including at least one field** and being indexed by indexes comprising n characters, the method comprising:” *Google* at Claim 1 |
| 1.b. **providing a list of the database fields, wherein the list includes a descriptor indicating a data category**;  *(cont.)*  1.b. **providing a list of the database fields, wherein the list includes a descriptor indicating a data category**; | **A.**  **US6189004**  “The method comprises accessing a schema description and a query interface description for the datamart. The schema description specifies a schema, which in turn, defines the relationships between the fact tables and dimension tables of the datamart. The **query interface description specifies the fields**, related to the schema description, that can be used in a query and the way in which results are to be presented to the user. The **fields correspond to columns and rows in the fact tables**.” *Rassen* at col. 2:66 through 3:8  “In a dimensional datamart, the data is typically organized as a star schema. At the **center of a standard star schema is a fact table** that contains measure data.” *Rassen* at col. 2:27-29  “The **fact column** 310 **lists all of the fact attributes within a single fact table** 304. The **fact column** 310 **includes** a cleanse flag, **a description**, a fact aggregate operator, a fact column key, a fact column name, a fact column number, a fact table key, and a physical type.” *Rassen* at col. 12:56-60  “6. The method of claim 1 wherein **the query mechanism description defines a query form, the query mechanism description** **defining a set of columns to be displayed on the query form, each column of the set of columns corresponding to a queryable field, the set of columns allowing an user to define a query for the datamart**.” *Rassen* at Claim 6  **B.**  **US6370518**  “20. A method for **displaying a record from a structured database on a screen, the record including at least one field and indexed by a desired index comprising a string of characters**, the method comprising: receiving a list of indexes of the structured database; displaying as many of the indexes sequentially as the screen can accommodate. . .” *Google* at Claim 20 |
| 1.c. **receiving a search selection for a database field on the provided list of the database fields**;  *(cont.)*  1.c. **receiving a search selection for a database field on the provided list of the database fields**; | **A.**  **US6189004**  “FIG. 13 illustrates a fact table window 1300 that is open on the order fact table 1310 definition. The fact data semantic 1310 is transactional/state like/force close/unjoined. The transactional/state like/force close/unjoined means that the invoice part of an order is transactional, the booking is state like, orders that are not otherwise dealt with, are closed out, and the data may become dirty and so it needs to be cleansed, thus, it is unjoined. This semantic type is described in detail in Appendix A. Note that the **user can select from many different types of fact table semantics**. The fact table window 1300 also shows the fact columns 1330 for the order fact table.” *Rassen* at col. 13:36-47  **B.**  **US6370518**  “According to one embodiment, zoom window 311 shows an initially chosen index 318 along with a field 319 of the record. This leaves a visual impression to a user that zoom window 311 is scrollable so as to allow scrolling to an index of interest. Preferably, zoom window 311 displays more than just an index. The **number of fields as well as which one or more fields of the associated record are caused to be displayed upon selecting one of the indexes**, however, does not affect the operation of the invention and largely depends on the actual implementation preference. For example, zoom window 311 may be chosen to display chosen index 318 and two fields, for example, a full name, a work phone number and a home phone number. Nevertheless, the number of the fields caused to be displayed in zoom window 311 determines the available space to display the rest of the indexes in screen 310.” *Google* at col. 6:29-44  “At 720, **the user determines if a desired index is among the indexes being displayed**. When the desired index is not displayed, the user enters a character query at 730. The counter is incremented by one every time the user enters a character query at 740. **The look-up process proceeds at** 750 **with the entered character and returns to** 720 **all indexes that have the entered character at the corresponding character location**. If the look-up process fails to identify any index in the database that matches the entered character and all precedent characters, an error message is displayed at 780.” *Google* at col. 10:9-18  “20. **A method for displaying a record from a structured database on a screen**, **the record including at least one field and indexed by a desired index comprising a string of characters, the method comprising: receiving a list of indexes of the structured database; displaying as many of the indexes sequentially as the screen can accommodate**. . .” *Google* at Claim 20 |
| 1.d. **determining a quantity of entries in the selected database field**; | **A.**  **US6189004**  “The **results of the query** are presented to the user **according the second set of commands**.” *Rassen* at col. 3:17-18  “The **aggregates correspond to pre-computed query results** for different types of queries. For example, an aggregate can be created for a query that asks for all sales, by region, by quarter. The corresponding **aggregate table would include a set of rows that have the results for this query** (e.g., each row includes the quarterly sales for each region).” *Rassen* at col. 5:6-12  **B.**  **US6370518**  “FIG. 7 illustrates a process flowchart of the present invention according to one embodiment and shall be understood in conjunction with FIG. 6. At 700, a portable device is powered on or the lookup process is activated. **A first portion of indexes from the database is displayed on the screen and a counter is set to zero** at 710. **The counter records the number of the character entries and can be used to indicate to a user how many characters have been entered**. At 720, the user determines if a desired index is among the indexes being displayed. When the desired index is not displayed, **the user enters a character query at** 730**. The counter is incremented by one every time the user enters a character query** at 740. The look-up process proceeds at 750 with the entered character and returns to 720 all indexes that have the entered character at the corresponding character location. If the look-up process fails to identify any index in the database that matches the entered character and all precedent characters, an error message is displayed at 780.” *Google* at col. 10:1-18 |
| 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 | **A.**  **US6189004**  “The fact table 304 defines the metadata 160 table describing all of the fact tables within a given constellation 302. The **attributes of the fact table** 304 **include** a build aggregates flag, a cleanse flag, a constellation key, a description, a fact table key, a fact table name, and **a truncate stage flag**. Each attribute corresponds to a column in the fact table 304 . . . The **truncate stage flag is used to indicate whether or not to truncate the fact staging table** on the next extraction.” *Rassen* at col. 12:38-55  “The job 402 is a top level object for controlling the work flow during the extraction and loading process 204. The job 402 includes a check databases field, a check tables field, a description, a label, an initial load flag, a job key, a job name, a log file width, a mail to on error, a mail to on success, and a truncate flag. The check databases field indicates whether or not an attempt should be made to log into all the data stores before executing the job. The check tables flag indicates whether or not to check for the existence of all the tables in the datamart 150 before executing the job. The description is for documenting the job (usually done by the consultant). The enabled flag indicates whether or not a particular job can be run. The initial load flag indicates whether or not to ignore all previous time stamped constraints when running a particular job. The job key is the primary key for the job table 402. The job name is the internal name of the job. The **log file width indicates how many characters wide to make rows in the log file output**. The mail to on error, and the mail to on success indicate where E-mail messages should be sent after failure or success of the particular job. The **truncate flag indicates whether or not to truncate any tables when running a job**.” *Rassen* at col. 18:1-21  **B.**  **US6370518**  “. . . **displaying on the display screen a reduced portion of the indexes, including the desired index, from the structured database, the reduced portion displays those of the indexes where each of the n characters in the indexes is respectively matched by one of the characters of the group of characters represented by the each of the symbols**.” *Google* at col. 3:12-18 |
| 1.f. **if the quantity does not exceed the specified amount, displaying contents of the database field**.  *(cont.)*  1.f. **if the quantity does not exceed the specified amount, displaying contents of the database field**. | **A.**  **US6189004**  “The **job** 402 is a top level object for controlling the work flow during the extraction and loading process 204. . . The log file width indicates how many characters wide to make rows in the log file output. The mail to on error, and the mail to on success indicate where E-mail messages should be sent after failure or success of the particular job. **The truncate flag indicates whether or not to truncate any tables when running a job**.” *Rassen* at col. 18:1-21  **B.**  **US6370518**  “1. A method for **displaying a record from a structured database on a display screen of an electronic device** having a reduced-size input interface, **the structured database including a plurality of records, the records including at least one field and being indexed by indexes comprising n characters**, the method comprising: displaying on the display screen a portion of indexes from the structured database . . .” *Google* at Claim 1 |