Abstract
This paper presents a Chinese query interface catered for casual Chinese users and does not follow the SQL standard syntactically or semantically, as SQL is mainly targeted for English users. SQL syntax reflects some flavor of English, but what is natural to English users may be quite unnatural to Chinese users. Yet, the designs of many existing Chinese interfaces still reflect a very strong SQL flavor. In this paper, a Chinese query language with constructs natural to Chinese users is defined. Also, a graphical query interface based on this query language is introduced. By making use of windows, the query interface allows complex queries to be decomposed into simpler sub-queries. This makes casual users as well as frequent users able to specify simple and even complex queries easily.

Keywords:
database interface, query interface, user interface, query language, query specification

1 Introduction
The widespread of information technology has brought the use of computers to a new era. In particular, the utilization of database management systems by the general public is becoming more and more popular. When database technology was first evolved, the access of computers was limited only to the professionals, where the training in the use of computers was part of their jobs. Today, people from all walks of life may have access to computers to obtain information from a database. An appropriate query interface for such casual users should require little learning, and must be natural to them. Numerous query interfaces have been proposed, but most of the approaches require the users to have certain background with computers, although this may not be true for most casual users.

For Chinese users, the problem is even worse. Although database technology has been existing for nearly three decades, its application has not gained much attention in China until recent years when high speed growth in computer usage occurred. As a result, interfaces today reflect the characteristics defined for English users. It is unreasonable to expect the general public in China be able to specify a query in English. Though there are some existing Chinese query interfaces, most are just direct translations from their English counterpart. This approach is certainly bound to fail because a query interface natural to an English user is not likely natural to a Chinese user because of the differences in the languages and cultures.

For example, Structured Query Language (SQL), the de facto industry standard query language, is supported by nearly all commercial relational database products today [1,2]. One may say that a faithful Chinese version of SQL is good enough for a Chinese query interface, as in approaches like [3,4]. However, when SQL was designed, it was aimed at providing a query language that is somewhat natural in English. Obviously, the construct "SELECT ... FROM ... WHERE" reflects a flavor of English rather than Chinese. Those with Chinese background will find the SELECT-FROM-WHERE sequence unfamiliar and difficult to grasp. The use of nesting in SQL further complicates the situation. This undoubtedly calls for the development of a query interface truly catered for Chinese users.

Our approach is to create a basic Chinese query interface which does not reflect SQL, but has the same expressive power of SQL. It should be natural to anyone knowing Chinese or only Chinese.

Further, with the development of windowing mechanism and graphics, more versatile query interfaces can be designed, as shown in [5, 6, 7, 8, 9, 10, 11, 12]. In our approach, complex queries can be decomposed into simpler sub-queries which later on can be recomposed to form the original complex query. This allows us to define a query interface easy to use for even casual users.

In the following sections of this paper, we shall first show the basic query language. After that, a casual graphical user interface based on this query language will be described.

2 Design Principles

2.1 Language Naturalness
One of our design objectives is not to force the users to memorize too many syntactical rules or structures of the query interface. As mentioned previously, most existing query interfaces are mainly catered for English users. For a Chinese user, the unnaturalness of an English interface makes it necessary to memorize the syntax and rules of the query interface reflecting SQL syntax or semantics.

Of course, using natural language is the most natural way for the user to specify a query. Some research projects heading in this way are [13, 14]. However, owing to the complexity of the natural language translator and the time-consuming process in translating, the use of unrestricted natural language query interface is difficult and cost-ineffective and in fact not possible. Our approach is to define a restricted natural language Chinese query language with a limited number of constructs for query specification such that each construct is natural to Chinese users. When used together the constructs are at least as powerful as SQL.

2.2 Divide and Conquer Strategy
It is common experience that when a person is asked to perform a complex task, it is easier for him or her to do it in a step by step manner. The person will first divide the task into smaller sub-tasks to make each task manageable for him or her. The completed sub-tasks are then brought together to form the final result. On the other hand, if the person is asked to perform the complex task as a whole, that is, the complex task must be viewed as a single unit, the chance that the complex task can be completed correctly, not to mention the extra time required, is much reduced.

We believe that the more natural way to specify a complex query is by the "Divide and Conquer" strategy [15, 16]. A complex query is first decomposed into sub-queries that can be easily managed by the user. Later on, the specified sub-queries are re-composed to form the final query. It is our observation that this strategy, though not explicitly taught, is widely used in database classes when students cannot specify a query in one single statement, but they can get the correct answer when the query is specified in multiple query statements. This strategy combines both declarative and procedural elements in the specification of a query and we believe it is the most suitable way for a query interface.

2.3 Unit of Operations
A concern for the "Divide and Conquer" strategy is that the query interface should not force the user to decompose a query when the user is comfortable to specify the query in one step or query statement. Thus our approach aims at allowing the user to decompose a query into any arbitrary number of sub-queries according to the user's background and ability.

To achieve this, we go back to the basic relational concept and operations of the relational model. Each of the relational operations proposed by Codd (e.g., projection, selection and join) has a very narrow scope and concise semantics. The basic operations in our query interface are resembled to the relational operations. In each sub-query, the user is allowed to specify multiple non-conflicting relational operations, and so the expressive power of each sub-query is quite large. For casual users, they may decompose a complex query into a number of sub-queries, while in each sub-query only one or two operations are specified. For the more experienced users, they may decompose the same query into only a few sub-queries, fully utilizing the expressive power of each sub-query to complete the construction of the whole query.

While conceptually this is the philosophy, operationally the interface reflects the Chinese query specification that we will define in the next section.

3 Query Specification in Chinese

3.1 Defining Elementary Operations
To make query specification natural to Chinese users, we should consider the natural data retrieval sequence of Chinese. Chinese queries can be concluded into a set of standard operations as follows:

Projection:
在<某表>中, 写出<某列>，将结果送入<某表>中。
(sequence: <table> → <column(s)> → put the result in <table>)

Restriction:
在<某表>中, 如果<某些列>的条件，写出满足这些条件的行，将结果送入<某表>中。
(sequence: <table> → <restriction condition(s)> → put the row(s) satisfying the condition(s) in <table>)

Join:
如果<某表>的<某些列>等于<某表>的<某些列>，写出满足这些条件的行，将结果送入<某表>中。
(sequence: <table> and <table> → <column(s) value> equal to <column(s) value> → put the row(s) satisfying the condition(s) in <table>)

Division:
在<某表>的<某些列>中，如果某<某些列>的条件，写出满足这些条件的行，将结果送入<某表>中。
(sequence: <column(s) of table> → <column(s) value> equal to <column(s) value> → put the row(s) satisfying the condition(s) in <table>)

Intersection:
Before we describe the details of the interface, we would introduce some common operations used in the interface. For each of these common operations, the name is shown as the label of the corresponding button in the screen. These operations are:

- **完成**: Current stage of input completed; proceed to the next stage.
- **取消**: Cancel the operation.
- **取消全部**: Cancel the whole query and return to the initial stage of interface.
- **执行**: Perform projection of attributes satisfying all assigned conditions.
- **输出**: Output the rows which satisfies the specified conditions.
- **全部添加**: Add all available attributes (or tables) to the selection list.
- **全部删除**: Delete all attributes (or tables) in the selection list.
- **添加条件**: Add a condition to the selection list.
- **删除条件**: Delete the specified condition in the selection list.

### 3.2 Selecting Required Tables

When the query session starts, the tables in the database are shown in Tables Window (TW) (Figure 3). TW lets the user select the tables they need in the query. This step is represented in (a) in Figures 1 and 2. This window is divided into two regions: <Tables available> and <Tables Selected>. The user may select the required tables (working tables) by any of the following methods:

1. **Click and Select**: Click on the button 标注 (select tables) first, and then click on the table(s) (in icon form) required. Click this button again for completing selection.
2. **Drag and Drop**: Click on the table icon without releasing the mouse button. Then, move the icon from the region 标注 to the region 已选择的表 (Drop action) and release the button (Drop action). Repeat the process until all required tables are selected.

These two methods of selection are the standard ways of doing selection in our interface. They are applicable to other operations as well (shown in later part of this paper).

After the user clicks on the button 输入完成 (complete), one of two menus will appear. If only one table is selected in TW, menu in Figure 4 will appear. Besides the common operations mentioned above, this menu provides the following option:

- **限制条件**: Specify the restriction condition(s).

On the other hand, Figure 5 will appear if more than two tables are selected. This menu is different from the previous one in that it provides one more option:
3.3 Performing Join
This window will be shown only if more than one tables
are selected. The join operation is performed in Join
Window (JW) (Figure 6).
The working tables are displayed in the upper section of
JW and they are represented by list-boxes where the
attributes of the tables are shown. Before performing
restriction or projection, the user should specify the join
conditions among the working tables. Since four tables
have been selected in TW, the syntax of Figure 2 is
applied. The join operation corresponds to the paths (2) to
(6). By applying the "Drag and Drop" operation, the user
can perform an equi-join on the tables. Other than using
"Drag and Drop" operation, the user can construct join
conditions by defining the three necessary components of
a join. They are:

\[
\begin{align*}
\text{属性} & \quad \text{运算符} & \quad \text{属性/值} \\
\langle \text{Field name} \rangle & \quad \langle \text{Operator} \rangle & \quad \langle \text{Field name/Value} \rangle
\end{align*}
\]
we can specify the condition(s) in the following manner:

\[
\begin{align*}
\text{属性} & \quad \to \quad \text{运算符} & \quad \to \quad \text{属性/值} \\
\langle \text{Attributename} \rangle & \quad \langle \text{Operator} \rangle & \quad \langle \text{Attribute name/Value} \rangle
\end{align*}
\]

For more condition(s)
These operations will be demonstrated in the later section.

3.4 Performing Restriction
The restriction conditions are constructed in Condition
Window (CW). This part of operations correspond to the
boxes (**) in Figure 1 and Figure 2. The Condition
Window (CW) is shown in Figure 7.
In the upper part of the window, all attributes of the
selected tables are listed and are available for the user to
select. In fact, the specification of the restriction is similar
to the process of specifying join conditions.

3.5 Performing Projection
Projection is performed in Projection Window (PW)
(Figure 8). PW is invoked when the button 寫列 is
clicked in JW or CW. The left part of PW is called
Available Fields Region (AFR) and it contains all the
working tables and their fields. To select fields, the user
may make use of the "Drag and Drop" principle or
initiating the "Add" operation by clicking on the button "輸出屬性". The fields clicked in AFR are inserted into
Selected Field List (SFL) in the right part of PW.

3.6 Completion of Query
The last stage is to initiate the execution of query by
clicking on the button 執行, or to store this query to the
database or the temporary area for later use by clicking on
the button 儲存. If the button 儲存 is clicked, the system
will prompt the user to enter a name for the query. Also,
the user will be asked where this query should be stored.
If the user chooses to store the constructed query to the
Temporary Area (TA) (Figure 9), the query would be
discarded after the user logout from the database system,
though it is available for re-use within the same session.

3.7 Other Basic Operations
It is clear the above elementary operations are not
sufficient to define all queries. A simple example is that
the action of grouping and selecting a special group
cannot be achieved by just using the join, restriction and
projection operations.
In our interface, this case is handled by the 數據分組
(grouping) operation. The user may click on the 數據分
組 button in CW and Group-by Window (GW) will appear
(Figure 10). The user may select the grouping fields
needed and define the group selection criteria in GW. In
the upper section of this window, the group-by attribute
can be selected by the "Drag and Drop" operation as
described earlier. The group selection criteria can be
constructed similarly as the join or restriction conditions.
In addition, this window also provides some special
operations such as 最多 (MAX), 最少 (MIN), 平均
(AVERAGE), 總和 (SUM) or 出現次數 (COUNT).

3.8 Composing and Decomposing a Complex Query
Earlier in the paper it is mentioned that almost all complex
queries are composed of a lot of simpler queries which can
be tackled easily. In our approach, each of these simple
queries can be constructed in our interface easily and their
results are stored in TA. These results can then be used to
re-compose the original complex query. The process of
decomposing and re-composing of a complex query would
be demonstrated in the next section.

4 Examples
In this section four examples would be described to
demonstrate the use of our interface. The sample database
used in the paper contains the following tables:
4.1 Example 1

Q1 列出女仕部門出售的 Sincere 產品。
(List the items supplied by Sincere and sold in men's department).
In this query, the operations performed would be:

<table>
<thead>
<tr>
<th>Tables selected</th>
<th>Join</th>
<th>Restriction</th>
<th>Projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>SALES (DEPT, ITEM, VOLUME)</td>
<td>DEPT (DNAME, FLOOR)</td>
<td>(SALES.DEPT IS EQUAL TO SUPPLY.DEPT)</td>
<td>(SUPPLY.ITEM)</td>
</tr>
<tr>
<td></td>
<td>ITEM (INAME, TYPE, COLOR)</td>
<td>(SUPPLIER.SNO IS EQUAL TO SUPPLY.SNO)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EMPLOYEE (ENO, ENAME, MGR, DEPT, SALARY, JOB-STATUS)</td>
<td>(SUPPLY.SUPPLIER IS EQUAL TO &quot;Sincere&quot;)</td>
<td></td>
</tr>
</tbody>
</table>

The join is performed by dragging the attribute 供應商編號 from 供應商 to the attribute 供應商 of 供應. After that, a line is used to connect the joining attributes to represent the existence of such a join.

On the other hand, we can enter the join condition by another method:
1. Click on the space in the lower section of JW under the left header 屬性. A list will appear and it contains all the attributes of the working tables for user's selection (Figure 11). In this example, the attribute 供應商.供應商編號 should be selected. It corresponds to the stage from (2) to (4) in Figure 2.
2. Click on the space in the lower section of JW under the header 运算符. A list containing all possible operators is shown for user's selection (Figure 12). The operator "等於 " is selected and this step represents the stage from (4) to (5) in Figure 2.
3. Similar to step 1, the user may specify the other join attribute under the right header 屬性 and this represents the stage from (5) to (6).

After that, a new blank line is appended under the specified join condition. To input more join conditions, the user may click in the new blank line and follow the above steps again. This corresponds to path (3) in Figure 2. The join conditions are connected by "and" though this is not explicitly shown. After inputting two conditions, the layout of the screen would be shown as Figure 13.

If user want to specify the restriction condition, he should click the button 輸入選擇條件 in Figure 13 to perform restriction.

In the CW, the user should input the following conditions:
(SALES.DEPT IS EQUAL TO "MEN'S") and (SUPPLY.SUPPLIER IS EQUAL TO "Sincere")

The user may now click on the button 寫列 to perform the projection operation in the Projection Window (PW). In our example, the only needed field for output is 銷售.產品 (SALES.ITEM). It may be selected by dragging the field 銷售.產品 from AFR to SFL (Figure 15). This stage corresponds to the path leading to box (b) in Figure 2.

4.2 Example 2

Q2 在員工數多於5的部門中，列出部門編號。
(For each department with more than 5 employees, list the department code)
In this query, the employees in each department should be treated as a group.
The operations performed would be:

Tables selected: 員工 (EMPLOYEE)
Group-by attribute: 員工.部門 (EMPLOYEE.DEPT)
Group-by restriction: 

員工.員工編號 的 出現次數 大於 5
COUNT(EMPLOYEE.ENO) IS GREATER THAN 5

Projection: 員工.部門

Following the convention of Chinese, the sequence of operations would be 
Tables selection → Group-by attribute → Group-by restriction → Projection

In fact, it is equal to the sequence of Figure 1 as follows:
(a) → 4 → 16 → 17 → 14 → 19 → (b)

In this case, since only one table is selected, menu in Figure 4 will appear and the user should click on the button 限制條件 to input restriction.

To start the group-by operation, the user should click on the button 數據分組 in CW. After that, the Group-by Window (GW) will appear and it allows user to select the group-by attributes and input the restriction condition of each group. In this example, the only attribute used for grouping is 員工.部門(EMPLOYEE.DEPT), and the only restriction criteria of each group is

員工.員工編號 的 出现次數 大於 5
COUNT(EMPLOYEE.ENO) IS GREATER THAN 5

To input the group-by attribute, the user can drag an attribute from the list under the title 可選擇的屬性 (Available attributes) and drop to the list under the title 已選擇的屬性 (Selected attributes).

In order to specify the group-by restriction, the user can specify the condition in the box under the title 分組條件. The input method is similar to the specifying of restriction condition and the result is shown in Figure 16.

After user has fill in the information in GW, button 輸入完成 should be clicked for returning to CW. In order to perform projection, button 寫列 in CW should be clicked. Similar to the steps specified in previous example; PW would be viewed as in Figure 17 after projection is being specified.

4.3 Example 3

Q3 在各部門中，列出員工平均薪金多於25000
或員工數多於 10 的部門編號。

(Show the department code for those departments with more than 10 employees or the average salary of employees in the department greater than 25000.)

This query can be considered as composed by two simpler queries. These queries are:

query 1: 員工平均薪金多於25000的部門。

(the average salary of employees in the department greater than 25000.)

query 2: 員工數多於 10 的部門編號。

(Departments with more than 10 employees.)

Similar to the procedures described in example 2, both of the queries can be constructed easily and stored in Temporary Area (TA) (Figure 18). Now the user should start a new query in TW. The user may drag and drop the two queries from TA to TW and treat them as normal tables (Figure 19).

After the user clicks on the button 輸入完成 (complete), the menu in Figure 5 is displayed. The user should select the button 集合運算 (set operations). A list of available set operators is then shown in another menu (Figure 20). Since it is a union operation for our example, the button 交 (merge) should be selected. After that, two new icons will appear in TA. One represents the result of the union and the other shows the query tree containing information of the new query.

By using this decomposition and re-composition techniques, the user only needs to deal with multiple simple queries, rather than having to face the whole query in one single step. As a result, this interface allows even casual user to construct very complex queries relatively easy in a few steps.

5 Summary and Conclusion

In this paper we have presented a basic Chinese query language that does not follow the syntax of SQL. It can fully be understood by any person knowing Chinese because we use a restricted subset of the Chinese natural language as constructs of the query language. The set of 66 test queries defined in the Appendix of [17] were used to test the power of our proposed query language and it is our claim that this Chinese query language is at least as expressive as SQL.

We have also defined a graphical end-user interface that allows users to define the Chinese query statements easily. Unlike existing Chinese interfaces which are mere Chinese versions of query interfaces for English users, this query interface does not force Chinese users to remember unnecessary syntax caused by the difference of Chinese and English styles and cultures. In fact, we believe that a graphical interface is also suitable for any relational users who are not thinking in a logical sequence as in the basic Chinese query language.

The method of defining complex queries in the interface is like the way people do things naturally, that is, decomposing a complex task into multiple manageable smaller tasks, and re-composing them to get the final result. The divide and conquer strategy allows the user to decompose a query into any arbitrary number of sub-queries depending on their expertise in using the interface. This allows both frequent and casual users to specify queries in the way which is most natural and convenient to them.
Currently, we are still doing research to make the basic Chinese query language even easier to specify. Also, a prototype of the proposed query interface is under implementation on top of Microsoft Windows.

References

If one table is selected
If two or more tables are selected