

Republic of Iraq  
Ministry of Higher Education  
And Scientific Research  
University of Baghdad  
College of Science  
Dept. of Computer Science



# COMPUTER ACRONYM DICTIONARY

A project  
Submitted to the Department of Computer Science,  
College of Science University of Baghdad in partial Fulfillment of  
Requirements for the Degree of B.Sc in Computer Science

**Prepared By**

Mohammad Hashim Khudair  
Suhaib Bader Hassan  
Dhuha Ja'far Khazal

**Supervised By**

Zayneb Raid Ahmed Al-Rubaie

Date of  
May-2009

## *Abstract*

*A computer Acronym Dictionary is helpful application software that gives the meaning of the word or acronym in computer science that belong to different categories like hardware, software, communications, network, etc... And display it in a form that enable the user to read and choose another word or add a new word to dictionary.*

*To build such application we need to deal with data base .A data base system is a computerized system, whose overall purpose is to store information and allow users to retrieve and update that information on demand. We use Microsoft Access to create our data base files.*

*AlsoVB6 programming language help us to connect between data base files that contents the meaning of the words with the dictionary interface.*

*Our Acronym dictionary can be used to translate from English to Arabic and verse versa.*

## Chapter One, Introduction

---

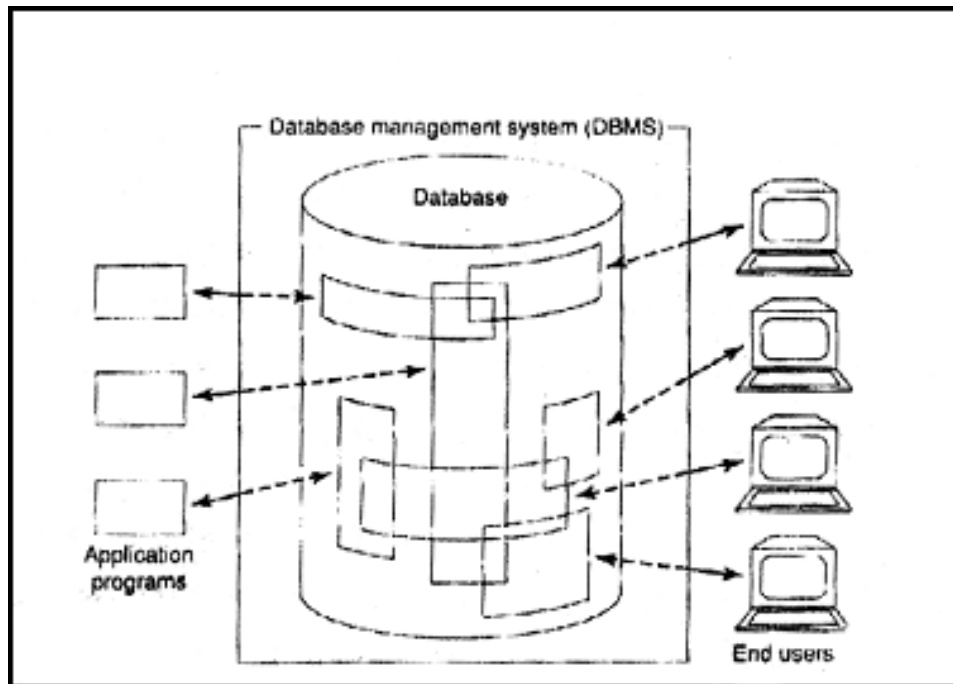
### 1.1 Introduction

The database technology has acquired a great of attention from the academic community and computer industry for their wide use and success. the past few decades witnessed developing and extending the database in response to the high progress of the applications that are used nowadays in most modern companies that need such technology to work with The use for such application and multimedia where essentially in these database and that withdraw scientist to search for techniques make it more and more efficient in handling these applications.

The increase in amount of all kinds available electronically has dramatically shown in recent years that encourage the researcher to find a way to minimize the size of space that data fit and reduce the search time in database.

### 1.2 What is a Database System?

A database system is basically a computerized record keeping system. in other words, it is a computerized system whose overall purpose is to store information and to allow users to retrieve and update that information on demand, the information in question can be anything that is of significance to the individual or organization concerned anything. in other words, that is needed to assist in the general process of running the business of that individual or organization incidentally. please note that we treat the terms data and information as synonyms in this project, some writers prefer to distinguish between the two using data to refer to what is actually stored in the database and information to refer to the meaning of that data as understood by some users. the distinction is clearly important, so important, that is seems preferable to make it explicit where appropriate instead of relying on a somewhat arbitrary differentiation between two essentially synonymous terms. Figure 1.1 is a simplified picture of a database system, as the figure shows such a system involves four major components data, hardware, software, and user we consider these four components briefly here.



*Figure 1.1 Simplified picture of a database system*

### 1.2.1 Data

Database systems are available on machines that range all the way from the smallest hand held or personal computers to the largest mainframes or clusters of mainframes. Needless to say the facilities provided by any given are determined to some extent by the size and power of the underlying machine. In particular, systems on large machines (large systems tend to be multi user) whereas those on smaller machines (small systems tend to be single user). A single user system is a system in

Which at most one user can access the database at any given time, a multi user system is a system in which many users can access the database at the same time. In fact, however the distinction is largely irrelevant so far as most users are concerned because it is precisely an objective of multi user system in general to allow each user to behave as if he or she were working with a single user system, instead the special problems of multi user system are primarily problems that are internal to the system not ones that are visible to the user now, it is convenient to assume, for the sake of

## **Chapter One, Introduction**

---

simplicity that the totality of data in the system is all stored in a single database and we will usually make that assumption here. However, there might be good reasons even in a small system why the data should be split across several distinct databases in general then the data in the database at least in a large system will be both integrated and shared. these two aspects data integration and data sharing represent a major advantage of database system in the large environment and data integration at least can be significant in the small environment as well. of course there are many additional advantages also to be discussed later even in the small environment, but first let us explain what we mean by the terms integrated and shared

- By integrated we mean the database can be thought of as a unification of several otherwise distinct files with any redundancy among those files partly or wholly eliminated. for example, a given database might contain both an EMPLOYEE file giving employee names and addresses, departments, salaries and so on, and an ENROLLMENT file representing the enrollment of employees in training courses (refer to fig 1.2). now suppose that in order to carry out the process of training course administration it is necessary to know the department for each enrolled student then there is clearly no need to include that information in the ENROLLMENT file because it can always be discovered by referring to the EMPLOYEE file instead.
- By shared we mean the database can be shared among different users, in the sense that different users can have access to the same data possibly even at the same time (concurrent access). such sharing concurrent or otherwise is partly, a consequence of the fact that the database is integrated in the example of fig 1.2. for instance, the department in the EMPLOYEE file would typically be shared by

users in the personnel department and users in the education department ( A database that is not shared in the foregoing sense is sometimes said to be personal or application specific ).

Another consequence of the foregoing facts that the database is integrated and shared is that any given user will typically be concerned only with some small portion of the total database moreover different users portion will overlap in various ways. in other words, a given database will be perceived by different users in many different ways in

## Chapter One, Introduction

---

fact even when two users share the same portion of the database their views of that portion might differ considerably at as detailed level.

EMPLOYEE	NAME	ADDRSS	DEPARTMEN	SALARY
ENROLLMEN	NAME	COURSE		

*Figure. 1.2. the EMPLOYEE and ENROLLMENT file*

### 1.2.2 Hardware

The hardware components of the system consist of:

- The secondary storage volumes typically magnetic disks that are used to hold the stored data together with the associated I/O devices (disk drives etc), device controllers, I/O channels and so forth.
- The hardware processor and associated main memory that are used to support the execution of the database system software (see the next subsection).

We does not concern itself very much with the hardware aspects of the system for the following reasons among others First those aspects from a major topic in their own right, second the problems encountered in this area are not peculiar to database system and third those problems have been very thoroughly investigated and documented elsewhere.

### 1.2.3 Software

Between the physical database itself that is the as physically stored and the users of the system is a layer of software known variously as the database manger or database server or most commonly the database management system (DBMS). All requests for access to the database are handled by the DBMS; the facilities for adding and removing files (or tables), retrieving data from and updating data in such files or tables and so on are all facilities provided by the DBMS. One general function provided by the DBMS is thus the shielding of database users from hardware level details (much as programming language system shield application programmers from hardware level details). In other words, the DBMS provides users with a perception of the database that

## Chapter One, Introduction

---

is elevated somewhat above the hardware level and it supports user operations such as the SQL operations in terms of that higher-level perception. A couple of further points

- The DBMS is easily the most important software component in the overall system but it is not the only one Others include utilities application development tools design aids report writers and (most significant) the transaction manager or TP monitor.
- The term DBMS is also used to refer generically to some particular product from some particular vendor, for example IBM's DB2 Universal Database product, the term DBMS instance is then sometimes used to refer to the particular copy of such a product that happens to be running at some particular computer installation as you will surely appreciate sometimes it is necessary to distinguish carefully between these two concepts.

That said you should be aware that people often use term database when they really mean DBMS (in either of the foregoing senses).

### 1.2.4 Users

We consider three broad (and some what overlapping) classes of users:

- First there are application programmers responsible for writing database application programs in some programming language such as COBOL, C++, Java or some higher level fourth generation language, Such programs access the database by issuing the appropriate request (typically an SQL statement) to the DBMS The programs themselves can be traditional batch application batch applications or they can be online application whose purpose is to allow an end user to access the database interactively (e.g. from an online workstation or terminal or a personal computer ) Most modern application are of the online variety
- Next there are end users who access the database interactively as described A given end user can access the database via one of the online application or he or she can use an interface provided as an integral part of the system Such vendor provided are also supported by means of online applications of course but those application are built in not user written Most system include at least one such

## **Chapter One, Introduction**

---

built in application called a query language processor by which the user can issue database requests such as SELECT and INSERT to the DBMS interactively, SQL is a typical of a database query language ( As an aside we remark that the term query language common though it is really a misnomer inasmuch as the verb to query suggests retrieval only whereas query languages usually not always provide update and other operators as well ).

Most system also provide additional built in interfaces in which end users do not issue explicit database requests such as SELECT and INSERT at all but instead operate by (e.g.) choosing items from a menu or filling in boxes on a form Such menu or forms driven interfaces tend to be easier to for people who do not have a formal training in IT (IT = information technology the abbreviation IS short for information system is also used with much the same meaning) By contrast command driven interfaces that is

Query languages do tend to require a certain amount of professional IT expertise though perhaps not much (obviously not as much as is needed to write an application program in a language like COBOL ) then again a command driven interface is likely to more flexible than a menu or forms driven one in that query languages typically include certain features that are not supported by those other interfaces .

### **1.3 What is a database?**

#### **1.3.1 Persistent data**

It is customary to refer to the data in a database as persistent (though it might not actually persist for very long). By persistent we mean, intuitively, that database, data not differs in kind from other more ephemeral data. such as input data, output data, work queues, software control blocks, SQL statements, intermediate results and more generally any data that is transient in nature. More precisely, we say that data in the database "persists" because once it has been accepted by the DBMS for entry into the database in the first place it can subsequently be removed from the database only by some explicit request to the DBMS not as a mere side effect of (e.g.) some program



## **Chapter One, Introduction**

---

completing execution this notion of persistence allows us to give a slightly more precise definition for the term database.

- A database is a collection of persistent data that is used by the application system of some given enterprise.

The term enterprise here is simply a convenient generic term for any reasonably self contained commercial, scientific, technical, or other organization. An enterprise might be a single individual (with a small personal database), or a complete corporation or similar large body (with a large shared database).

### **1.3.2 Entities and Relationships**

We now consider the example of a manufacturing company ("know ware Inc"). Such an enterprise will typically wish to record information about the projects in the on hand; the parts that are used in those projects; the suppliers who are under contract to supply those parts; the warehouses in which those parts are stored; the employees who work on those projects and so on. Projects, parts, suppliers, and so on. thus constitute the basic entities about which know ware Inc, needs to record information ( the term entity is commonly used database circles to mean any distinguishable object that is to be represented in the database ). Refer to figure 1.3.

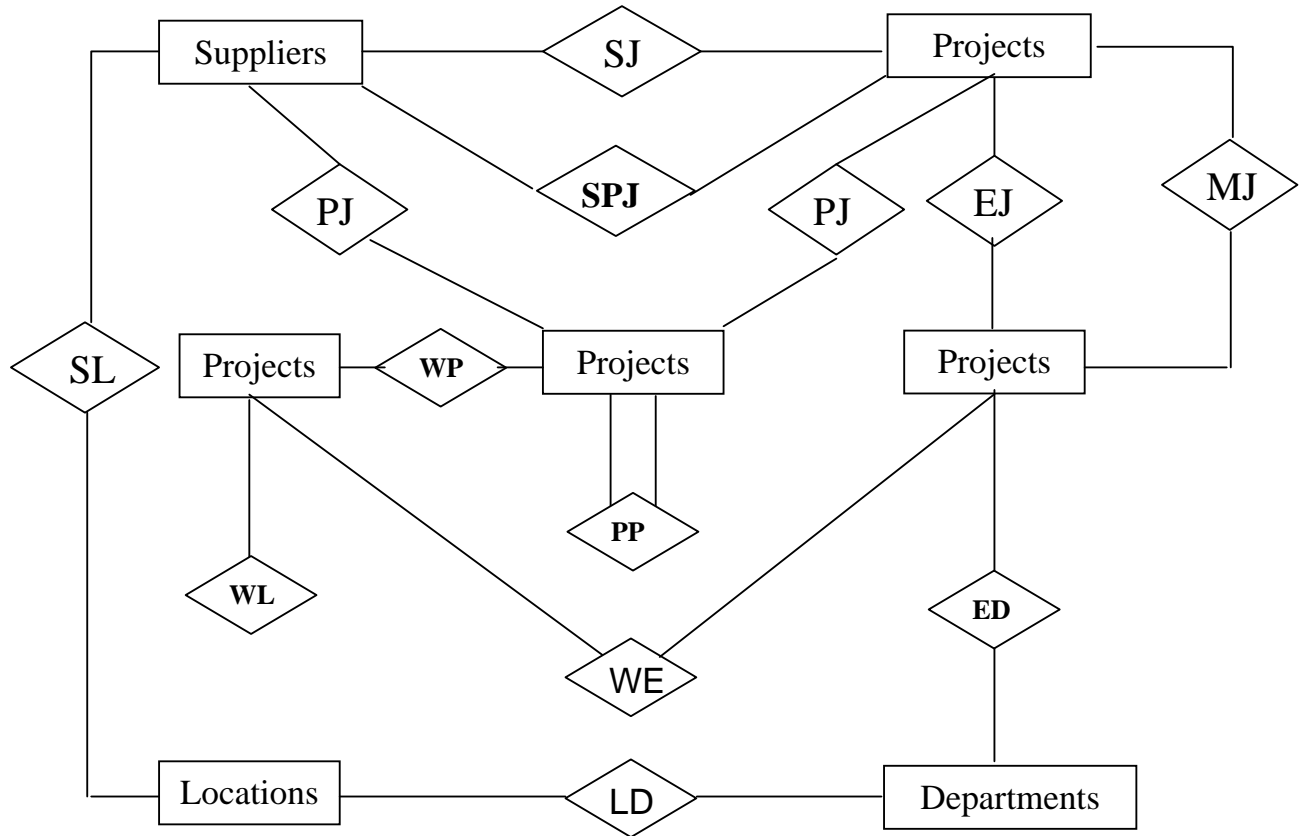


Figure 1.3 entity / relationship (E / R) diagram for know ware Inc

In addition to basic entities themselves (suppliers parts and so on in the example) there will also, the relationships linking those basic entities together, such relationships are represented by diamonds and connecting lines in figure 1.3. For example, there is a relationship (SP or shipments) between suppliers and parts: Each supplier supplies (more accurately, each supplier supplies certain kinds of parts, each kind of part is supplied by certain suppliers). Similarly; parts are used in projects, and conversely projects use part (relationship PJ); parts are stored in warehouses and warehouses store parts (relationship WP) and so on note that these relationship are all bi-directional that is they can be traversed in either direction, for example relationship SP between suppliers and parts can be used to answer both of following queries:

- Given a supplier get the parts supplied by that supplier.
- Given a part get the suppliers who supply that part.

## Chapter One, Introduction

---

The significant point about this relationship (and all of the others illustrated in the figure) is that they are just as much a part of the data as are the basic entities they must therefore be represented in the database just like the basic entities.

We note in passing that figure 1.3 is an example of what is called (for obvious reasons) an entity / relationship diagram (E / R diagram for short).

Figure 1.3 also illustrates a number of other important points.

Although most of the relationships in that figure involve two entity types, that is, they are binary relationships it is by no means the case that all relationships are binary in this sense in the example there is one relationship (SPJ) involving three entity types (suppliers parts and projects) a ternary relationship the intended interpretation is that certain suppliers supply certain parts to certain projects, Note carefully that this ternary relationship (“suppliers supply parts to projects”) is not equivalent.

In general, to the combination of the three binary relationships “suppliers supply” parts. Parts are used in projects; and supplied by suppliers, for example, the statement that :

- Smith supplies monkey wrenches to the Manhattan project, tells us more than the following three statements.
- Smith supplies monkey wrenches.
- Monkey wrenches are used in the Manhattan project.
- The Manhattan project is supplied by Smith.

### 1.4 Why database?

Why use a database system? What are the advantages? To some extent the answer to these questions depends on whether the system in question is single-or multi-user (or rather, to be more accurate, there are numerous additional advantages in the multi-user case). We consider the single-user case first.

Consider a simple stock as example, which we can regard as illustrative of the single-user case. Now, that particular database is so small and so simple that the advantages might not be all that obvious. But imagine a similar database for a large restaurant. with a stock of perhaps thousand of bottles and very frequent changes to that stock. The

## **Chapter One, Introduction**

---

advantages of a database system over traditional, paper-based methods of record keeping are perhaps easier to see in these cases. Here are some of them:

- Compactness: There is no need for possibly voluminous paper files.
- Speed: The machine can retrieve and update data far faster than a human can in particular, ad hoc, spur-of-the-moment queries, can be answered quickly without any need for time –consuming manual or visual searches.
- Less drudgery: much of the sheer tedium of maintaining files by hand are eliminated mechanical tasks are always better done by machines.
- Currency: Accurate up-to-date information is available on demand at any time.
- Protection: The data can be better protected against unintentional loss and unlawful access.

The foregoing benefits apply with even more force in a multi-user environment, where the database is likely to be much larger and more complex than in the single-user case. However, there is one overriding additional advantage in such an environment: The database system provides the enterprise with centralized control of its data (which,

As you should realize by now, is one of its most valuable assets). Such a situation contrasts sharply with that found in an enterprise without a database system; where typically each application has its own private files-quite often its own private and disks too-so that is widely dispersed and difficult to control in any systematic way.

### **1.5 Data Administration and Database Administration**

We elaborate briefly on this concept of centralized control the concept implies that there will be some identifiable person in the enterprise who has this central responsibility for the data and that person is the data administrator (DA for short), it is imperative that there should be some person who understands that data, and the needs of the enterprise with respect to that data, at a senior management level. The data administrator is that person, Thus, it is the data administrator's job to decide what data should be stored in the database in the first place and to establish policies for maintaining and dealing with that data once it has been stored. An example of such a

## **Chapter One, Introduction**

---

policy might be one that dictates who can perform what operations on what data in what data in what circumstances in other words a data security policy.

Note carefully that the data administrator is a manager not a technician (although he or she certainly does need to have some broad appreciation of the capabilities of database systems at a technical level) the technical

person responsible for implementing the data administrator's decisions is the database administrator (DBA for short), the DBA unlike the data administrator is thus an information technology (IT) professional. the job of the DBA is to create the actual database and to put in place the technical controls needed to enforce the various policy decisions made by the data administrator. The DBA is also responsible for ensuring that the system operates with adequate performance and for providing a variety of other technical services. The DBA will typically have a staff of system programmers and other technical assistants (i.e. the DBA function will typically be performed in practice by a team of people. Not just by one person). For simplicity. however, it is convenient to assume that the DBA is indeed a single individual.

### **1.6 Benefits of the Database Approach**

In this subsection we identify some more specific advantages that accrue from the centralized control

- The data can be shared.

Sharing means not only that existing application can share the data in the database but also that new application can be developed to operate against that same data. in other words, it might be possible to satisfy the data requirements of new applications without having to add new data to the database.

- Redundancy can be reduced.

In non-database system each application has its own private files. This fact can often lead to considerable redundancy in stored data. With resultant waste in storage space. For example, a personnel application and an education records application might both own a file that includes department information for employees. However, those two files can be integrated, and the redundancy eliminated as long as the data administrator is

## **Chapter One, Introduction**

---

aware of the data requirements for both applications that is, as long as the enterprise has the necessary overall control.

Note: We do not mean to say that all redundancy can or necessarily should be eliminated, sometime there sound business or technical reasons for maintaining several distinct copies of the same data. However, we do mean that any such redundancy should be carefully controlled: that is, the DBMS should be aware of it, If it exists, and should assume responsibility for “propagating updates”(see the point immediately following).

- Inconsistency can be avoided (to some extent).

This is really a corollary of the previous point Suppose a given fact about the real world say the fact that employee E3 works in department D8 – is represented by two distinct entries in the database. Suppose also that the DBMS is not aware of this duplication (i.e. the redundancy is not controlled). Then there will necessarily be occasions on which the two entries will not agree: namely, when one of the two has been up data and

the other not. At such the database is said to be inconsistent. Clearly, a database that is in an inconsistent state is capable of supplying incorrect or contradictory information to its user.

Of course, if the given fact is represented by a single entry (i.e., if the redundancy is remove), then such an inconsistency cannot occur. Alternatively, if the redundancy is not removed but is controlled (by making it known to the DBMS), then the DBMS can guarantee that data base is never inconsistent as seen by the user, by ensuring that any change made to either of the two entries is automatically applied to the other one as well. This process is known as propagating updates.

- Transaction support can be provided.

A transaction is a logical unit of work (more precisely, a logical unit of database work), typically involving several database operations-in particular, several update operation.

The standard example involves the transfer of a cash amount from one account A to another account B. Clearly two updates are required here, one to with draw the cash from account A and the other to deposit it to account B. If the user has made the cash from account and the other to deposit it to account B. If the user has made the two

## **Chapter One, Introduction**

---

update part of them are done or neither is— even if, for example, the system fails (say because of a power outage) halfway through the process.

- Integrity can be maintained.

The problem of integrity is the problem of ensuring (as far as possible) that the data in the database is correct, Inconsistency between two entries that purport to represent the same fact is an example of lack of integrity. of course, this particular problem can arise only if redundancy exists in the stored data. Even if there is no redundancy, however, the database might still contain incorrect information. for example, an employee might be shown as having worked 400 hours in the week instead of 40, or as belonging to a department that does not exist. Centralized control of the database can help in avoiding such problems insofar as they can be avoided-by permitting the database administrator to define, and the DBA to implement. Integrity constraints to be checked when update operations are performed.

It is worth pointing out that data integrity is even more important in a database system than it is in a “private files” environment. Precisely, because the data is shared. For without appropriate controls it would be possible for one user to update the database incorrectly, thereby generating bad and so “infecting “ other innocent users of that data. It should also be mentioned that most database products are still quite weak in their support for integrity constraints (though there have been some recent improvements in this area). This fact is unfortunate, given that integrity constraints are both fundamental and crucially important—much more so than is usually realized.

### 2.1 DATABASE ACCESS

Sharing expensive network printers and disk storage across mixed environments helps companies save money. But what about making use of the important business data residing on different systems throughout the organization? Again, rather than moving the data, often a difficult and costly endeavor, organizations need the ability to access this information from key enterprise business applications. Somehow, they need to make the data sitting on a UNIX workstation or IBM DB2 database accessible to the business applications running on PCs. And they need to make this happen in a way that is transparent to the PC user. Providing universal access to data is vital to developing powerful distributed solutions for running the business.

Two fundamental components of Microsoft's universal data access strategy are the Open Database Connectivity (ODBC) standard and the OLE DB standard. ODBC provides a unified way to access relational data from heterogeneous systems; any application that supports ODBC can access information stored in any database that houses relational data. For example, a Microsoft Excel spreadsheet, Microsoft Access database, or any other ODBC-enabled application could pull and use data from an ORACLE, SYBASE, Informix, or any other UNIX relational database without installing any software on the UNIX system. A finance specialist using Microsoft Excel to track quarterly shipments of sweatshirts could pull inventory data from an Oracle database running on UNIX to generate a report.

With ODBC, developers do not need to write separate client/server applications to access individual UNIX databases; by simply supporting ODBC, a single application can access a variety of UNIX relational or host databases.

Exchanging data—not just accessing it—is also possible among heterogeneous databases, allowing organizations to deploy Windows NT platforms in distributed business units while keeping UNIX workstations, midrange systems, or mainframes in a centralized location. For example, through heterogeneous replication, a SQL Server™ 6.5 database running on Windows NT can automatically send data to an Informix database running on a UNIX system.

If all of your data were stored in relational databases, integration could largely be solved with ODBC alone. But most data is non-relational, or un-structured.



## **Chapter Two, Database in Visual Basic**

---

For example, audio and video clips e-mail messages, and Notes documents represent a few of the types of unstructured data stored in various formats throughout most organizations. To simplify integration of unstructured data across the enterprise, Microsoft offers OLE DB. Analogous to ODBC in the relational world, OLE DB is a set of OLE interfaces that provides applications with uniform access to unstructured data regardless of type or location on the network. OLE DB is technology for software vendors whose products either use or provide unstructured data. Developers can write applications that connect to any OLE DB data provider—whether a file system, Oracle database, Excel spreadsheet or DB2 database—and let PC users running Windows desktop applications share and manipulate data stored there.

To tap the vast potential of the Internet and intranet to run business applications, organizations need access to information residing in UNIX and host databases from a Web browser. With the Internet Information Server running on Windows NT, organizations can build Web sites that access information from any database that supports ODBC, such as Sybase or Oracle. The Internet Information Server supports ODBC through an Internet Database Connector, enabling developers to build Web pages that make dynamic queries to UNIX, IBM, or any other ODBC-compliant databases.

Developers also can give browser users the ability to alter, add, and delete data they have queried and retrieved from UNIX or host databases; end users are not restricted to staring at static HTML results tables. Microsoft's Advanced Data Connector (ADC), introduced in December 1996, is a high-performance, Web-based technology that enables developers to build Internet and intranet applications with database connectivity and corporate data publishing capabilities. Users can access and update data from any ODBC-compliant database management system.

### **2.2 DISTRIBUTED, HETEROGENEOUS BUSINESS APPLICATIONS**

Microsoft's goal is to offer customers the most extensive interoperability possible between UNIX and Windows environments. Beyond file transfer, printer-sharing and universal data access, Microsoft strives to let organizations build distributed, heterogeneous business applications that integrate the various data, languages, and systems in which they have invested throughout the enterprise. This

## Chapter Two, Database in Visual Basic

---

capability is delivered in two primary ways: 1) by providing customers a set of tools and technologies for powerful, cross-platform application development and 2) by supporting UNIX desktop clients in BackOffice server applications, such as the Microsoft Exchange Server messaging platform and the Microsoft Proxy Server.

### 2.3 ADODB

The Object Browser is a powerful tool for learning about object models. This tool will become especially important to Access 2000 developers because of the many significant new object model innovations. For example, Access 2000 introduces a new data access development language—ActiveX Data Objects (ADO)—that will eventually make Data Access Objects (DAO) obsolete. There are at least three object models underlying the Access 2000 implementation of ADO.

Recall that you will need to set references to many libraries. Use the Tools | References command to verify or create links to ADO type libraries. Open the Object Browser by clicking its button on the standard VBE toolbar. You can also present the browser with the View menu or by clicking 2.

One of the ADO libraries is named ADODB. Select this from the drop-down list at the top of the Object Browser. Notice that this action changes the contents of the Classes and Members lists in the browser. Figure 10 shows the Recordset class selected in the Classes list with a mix of properties, methods, and events appearing in the Members list. Select a member entry and click the ? button for detailed help on that topic.

The Object Browser can also search for specific classes and members with its search engine. Use the second drop-down box to specify a search criterion. This will sometimes develop result sets in several libraries. The Object Browser enables you to examine these separately.

### 2.4 RECORDSET SCRIPT OBJECT

The Recordset object makes it easier to manipulate data by using its properties, methods, and events. Data-bound objects, which can be created by design-time controls, are bound to the Recordset object. It is the Recordset object that controls the flow of data. Therefore, you must create a Recordset object before you create other script objects if you want to bind data.

## Chapter Two, Database in Visual Basic

---

At design time, you can use the Recordset design-time control to create a Recordset object. The Recordset control has property pages that make it easier to control the behavior of the Recordset object.

When you are using design-time controls, there are two ways to add a Recordset control to a page. You can:

Drag a Recordset control from the Toolbox's Design-Time Controls pane onto a page and then specify the data connection and other properties.

In the Project Explorer, drag a data command from the Data-Environment node onto a page. The Recordset control's properties are automatically set to the DE object (Data-Environment).

To move the cursor position within the recordset, call the methods `move`, `moveFirst`, `movePrevious`, `moveNext`, `moveLast`, `moveAbsolute`, `getBookmark`, and `setBookmark`.

Use the `BOF` and `EOF` properties to verify if the current record location is valid. To modify the recordset, call the methods `updateRecord`, `cancelUpdate`, `addRecord`, and `deleteRecord`.

To return the number of records in the recordset, call the `getCount` method, when opening and closing a recordset, call the `open`, `close`, and `isOpen` methods.

The Recordset object has a child object named `fields` that is a fields collection. Use the `fields` object to return values and objects that are related to the Recordset object's fields.

### 2.5 VIEWING DATA

In Microsoft® Visual InterDev™, you can display data on your Web pages using data-bound design-time controls. The new data environment makes this process easy by giving you the ability to create and manage all your data-bound controls from one location.

First, you make a data connection to an existing database. The data connection appears in the data environment (Data-Environment) folder in your Web project. You can then easily add controls bound to this connection to an ASP or HTML page by creating Command objects in the data environment and dragging them to the page. Visual InterDev creates data-bound controls that display the data from the database.

## **Chapter Two, Database in Visual Basic**

---

You can also drag data-bound controls from the Toolbox to your page to display data. For example, you can create a data connection and add a Recordset design-time control based on this connection to an ASP or HTML page. A number of other controls allow you to display data from the recordset defined by a Recordset control in different ways: text boxes, labels, list boxes, check boxes, option groups, and so on.

For even more flexibility in presenting your data, you can take advantage of the Grid design-time control. You can use this control to display data from multiple records in a grid format.

### **2.6 TABLE AND DATABASE DESIGN**

Before you enter any information in Access, you have to create a table somewhere. A table is the foundation for all queries, forms, and reports. The container for all these objects is the database.

In this part, you find tasks showing how to create your database file and the tables that belong to the database. You follow steps for creating the basic building blocks of a table--fields. In addition, you find reference information for setting the data type, format, default value, size, and other properties of fields. To speed up data retrieval and link to other tables, you can also index the fields, set their primary keys, and create relationships between tables.

### **2.7 DATA: BLANKS, NULLS, AND ZERO-LENGTH STRINGS**

When entering information into your database records, you can leave data out of a field if it is not available. However, a blank field can mean more than one thing. It might signify that you don't know what the field's value is or that a value for the field doesn't exist. You can leave a field blank if you don't know what the value is (the value in this field is actually called a Null value). If you set the Zero-length property, you can enter "" (two quotes) in the field to indicate there is no value.

### **2.8 DATA: FORMAT**

In addition to decimal places (see "Decimal Places"), you can change the format of a number so that all numbers look consistent in your table. Numeric formats

## **Chapter Two, Database in Visual Basic**

---

include dollar signs, percent signs, and commas. Date formats include spelling the months or using numbers for the month, and how many digits to use for the day and year. Text formats include capitalization.

Queries, forms, and reports also enable you to change the format places. On these objects you can change Format by right-clicking the field or control in Design View and making the change on the Property sheet in the forms.

### **2.9 DATA TYPES: CHANGING**

When you create a table, you specify the data type for each field in the Table Design View. Text, Number, Date/Time, Currency, and Yes/No data types are self-explanatory. Auto Number, formerly called Counter in previous versions of Access, increments each record by one. While Text fields can only contain up to 255 characters, Memo can include over 65,000 characters. OLE Objects enable you to insert graphics, sounds, and other data types. The Lookup data type will enable you to choose from a list of options. (See "Lookup Columns: Create with Wizard.") The Hyperlink data type enables you to launch an Access object (a table, form, query, and so on), another file in another application, or go to a Web site. (See "Hyperlinks: Create Field.")

### **2.10 DATA TYPES: SETTING DEFAULTS**

When you first create a field in Table Design View, the field is automatically set to text and the default text field size is 50. If most of your fields are not text or are a different size, you can change these defaults.

### **2.11 DATABASE: CREATE BLANK**

A blank database is a database file that contains no objects or data. It is an empty shell that you will use to add new tables, queries, forms, and reports. If you want Access to create some of your tables, forms, and other objects, see "Database: Create New with a Wizard."

### **2.12 PROVIDER PROPERTY in VB**

Indicates the name of the provider for a Connection object. Sets or returns a String value that indicates the provider name. Use the Provider property to set or return

## **Chapter Two, Database in Visual Basic**

---

the name of the provider for a connection. This property can also be set by the contents of the Connection-String property or the Connection-String argument of the Open method; however, specifying a provider in more than one place while calling the Open method can have unpredictable results. If no provider is specified, the property will default to MSDASQL (Microsoft OLE DB Provider for ODBC).

The Provider property is read/write when the connection is closed and read-only when it is open. The setting does not take effect until you either open the Connection object or access the Properties collection of the Connection object. If the setting is not valid, an error occurs.

### 3.1 Main Menu

Figure 3.1 represents the start up form of our project

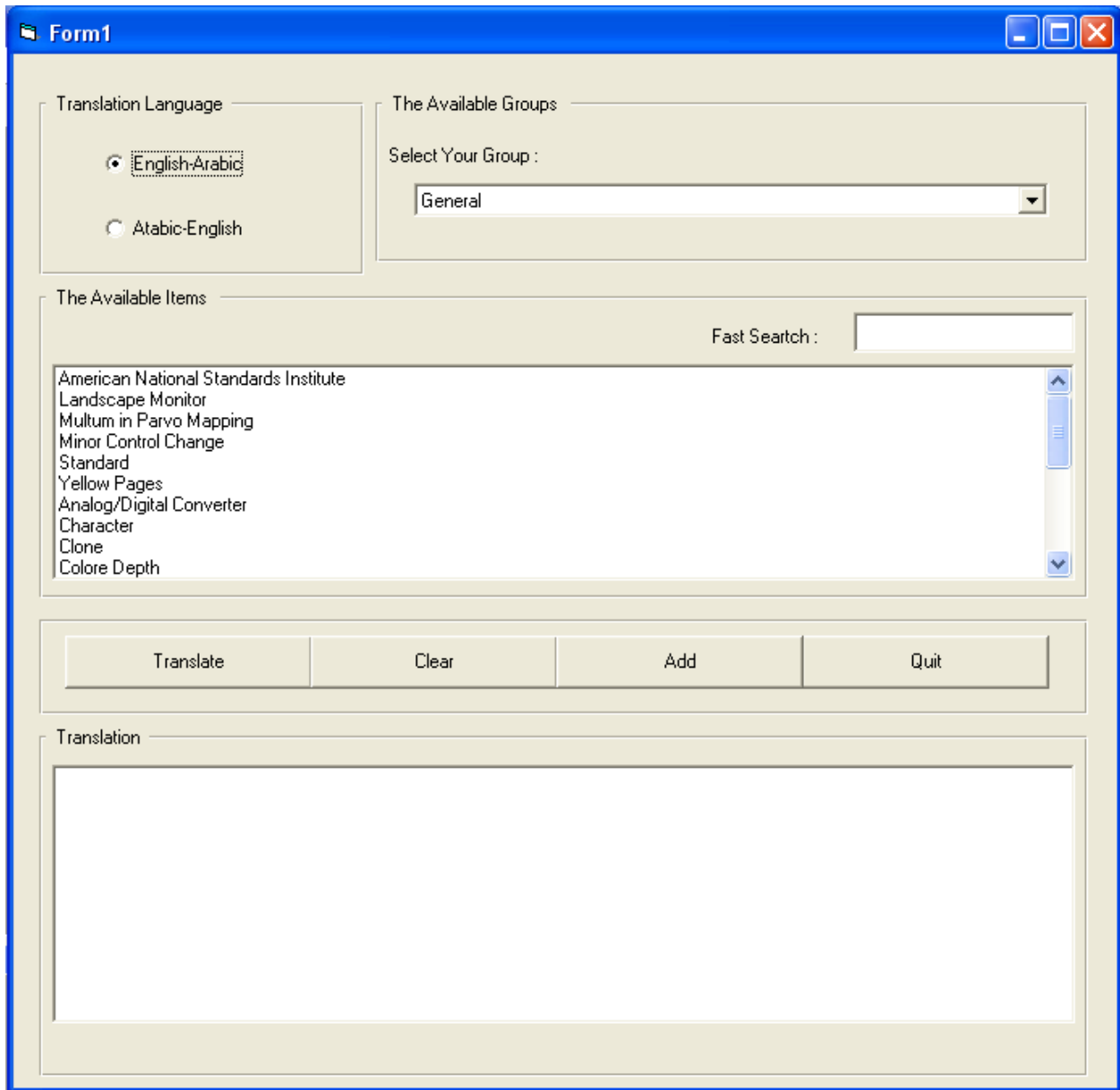


Figure 3.1 start up form

### 3.2 Frames

## Chapter Three, Practical Part

There are five frames in start up form:-

### 3.2.1 Translation Language

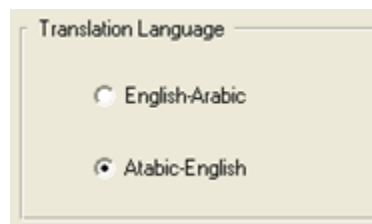
This frame contain two options button, the first one its caption is (English-Arabic)

When we choose it the dictionary translates from English to Arabic. See figure3.2



**Figure- 3.2**

The second button caption is (Arabic-English) when we choose it the dictionary translates from Arabic to English. See figure 3.3



**Figure 3.3**

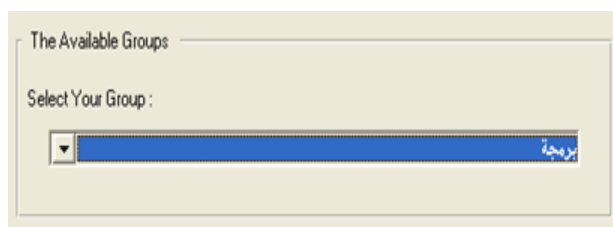
### 3.2.2 The Available Groups



## Chapter Three, Practical Part

The second frame contains a label (select group) and a combo box for the name of groups

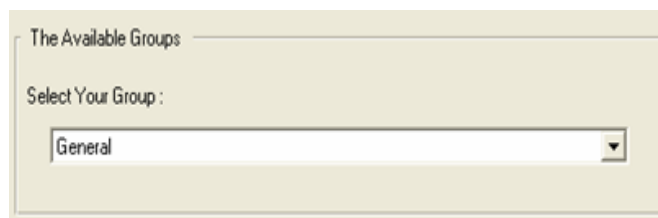
In Arabic:



**Figure 3.4**

---

In English:



**Figure 3.5**

### 3.2.3 The Available item

The third frame contains:-

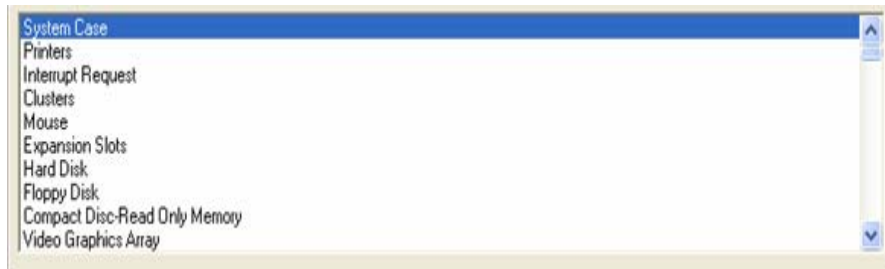
- Label for fast search and text box to enter the letter or entire word to search the meaning of it in our dictionary.



**Figure 3.6**

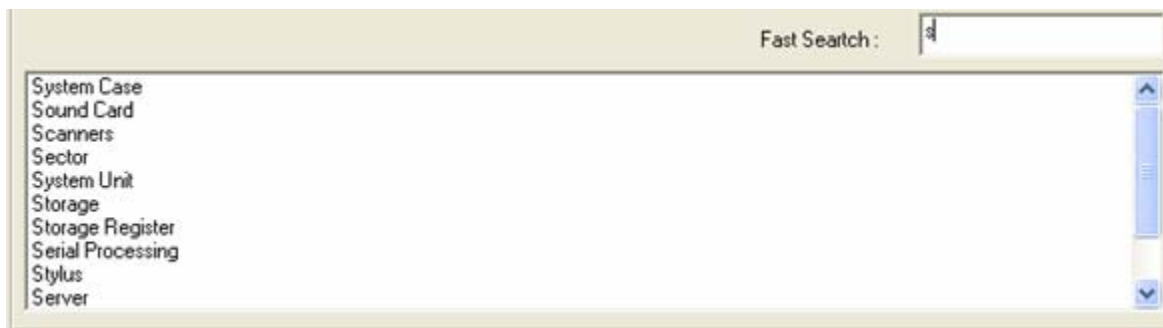
## Chapter Three, Practical Part

- List box explores the items under the selected group or the items that begins with the similar letter of searched word. See figure 3.7



**Figure 3.7**

For example when we write the letter 's',all words begin with this letter will appear



**Figure 3.8**

### 3.2.4 Commands

The fourth frame contains four command buttons:-



**Figure 3.9**

- **Translate**

When we click on this command the meaning of the word we wrote in text box will be appear in the list bellow

Note: we can choose the word randomly without write a letter on search.

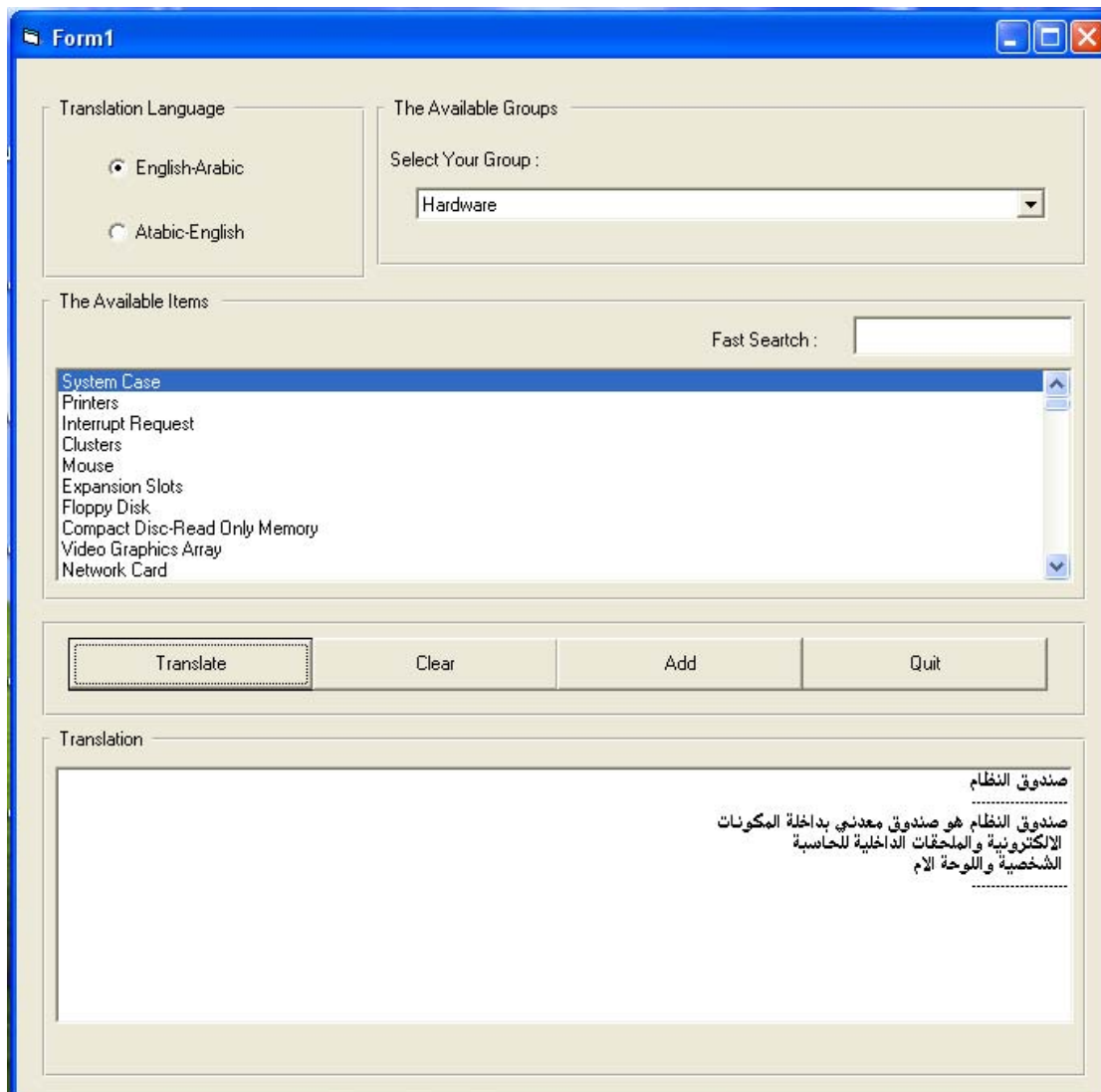


Figure 3.10

- **Clear**

This command clears the list below that displays the meaning of words.

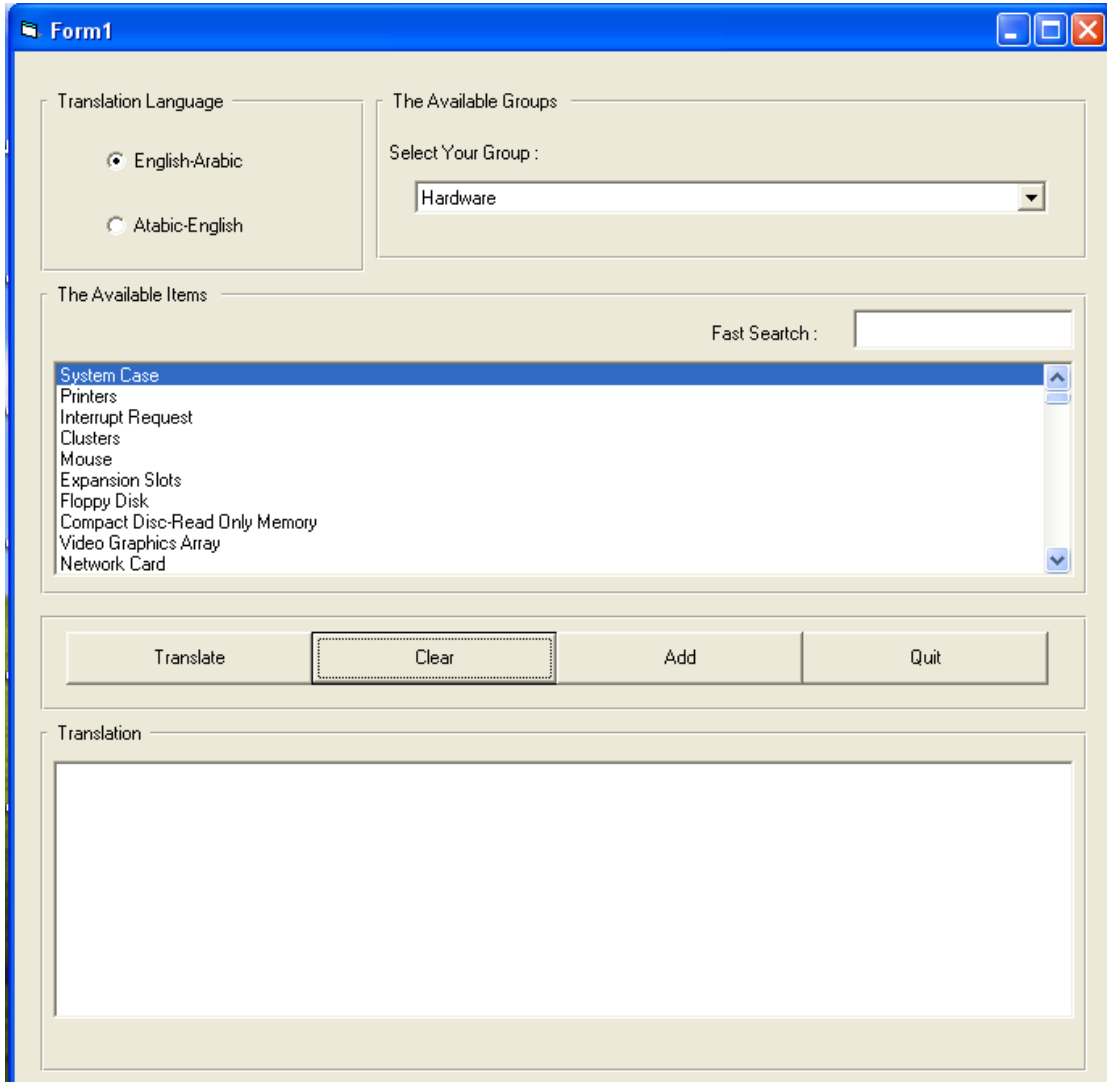


Figure 3.11

## Chapter Three, Practical Part

- **Add**

If we want to add a new Acronym to the dictionary we click on this command, the new form will be appear, see figure 3.12 this form contains three frames, the *first and second* frames was mentioned in figures (3.2, 3.3, 3.5).

The *third* frame called (**add your translation**), this frame contain four text box with their labels for adding the acronym and the meaning of it, so we can add the explanation of this acronym. When we finish writing it, we click on command button (**add to database**) to add the new information in the table of database, then click ok.

The screenshot shows a dialog box titled "Adding Items to the Existed Groups". It is divided into three main sections. The top-left section, "Translation Language", contains two radio buttons: "English-Arabic" (which is selected) and "Atabic-English". The top-right section, "The Available Groups", features a dropdown menu labeled "Select Your Group:" with "General" selected. The bottom section, "Add Your Translation", contains four text boxes: "Apprivation:" with the value "cpu", "English Meaning:" with "control process unit", "Arabic Meaning:" with "وحدة المعالجة المركزية", and "Arabic Explain:" with "تقوم بمعالجة وتنفيذ جميع البرامج". At the bottom right of the dialog is an "Add to Database" button, and at the bottom center is an "OK" button.

Figure 3.12

- **Quit**

When we click on this command the program is ended.



## Chapter four: Future work

### Future work:

1. An idea for future work is to make our dictionary is general dictionary for all language type for example: from English to Franc, from France to Arabic and verse versa.
2. Make our dictionary as learning package or an entertainment and useful application, by adding vocabulary test game or intersection word game depend on the data base of acronym dictionary.
3. For more explanation to computer acronym we can add a picture with the meaning of each one.