Design of Intelligent Database System for Vehicle Fault Diagnosis

By

Saeed Mohammad Saeed Tayem

Supervisor

Dr. Hussein H. Owaied

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Master Degree in

Computer Science

Faculty of Information Technology

Middle East University

2011

AUTHORIZATION

إقرار تفويض

أنا سعيد محمد تيم أفوض جامعة الشرق الأوسط بتزويد نسخ من رسالتي للمكتبات أو المؤسسات أو الهيئات أو الأفراد عند طلبها.

التوقيع:

التاريخ: ١٥/٨/٢٠١١

Authorization statement

I, Saeed Mohammad Saeed Tayem, authorize the Middle East University to supply a copy of my thesis to libraries, establishments or individuals upon their request.

Signature:

Date: 15/8/2011

Middle East University

Examination Committee Decision

This is to certify that the thesis entitled "Design of Intelligent Database System for Vehicle Fault Diagnosis" was successfully defended and approved on August 1, 2011

Examination Committee Members	Signature
Dr. Hussein H. Owaied Associate Professor, Department of Computer Science (Middle East University)	....
Dr. Ashraf Bany Mohammed Assistant Professor, Department of Computer Information System (Middle East University)	....
Prof. Dr. Musbah J Aqel Professor, Department of Computer Engineering (Applied Science Private University)	M. Aqel ....

DECLARATION

I do my hereby declare the present research work has been carried out by me under the supervision of Dr. Hussein H. Owaied, and this work has not been submitted elsewhere for any other degree, fellowship or any other similar title.

Signature:

Date: 15/8/2011

Saeed Mohammad Saeed Tayem

Department of Computer Science

Faculty of Information Technology

Middle East University

ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. Hussein H. Owaied, for his support, encouragement, proofreading of thesis drafts, and helping me throughout my thesis, and so putting me on the right track of Artificial Intelligence. I thank the Information Technology Faculty members at the Middle East University for Graduate Studies. I thank my father and my mother for their continuous support during my study.

DEDICATION

Almighty Allah says “And remember! your Lord caused to be declared (publicly): "If ye are grateful, I will add more (favours) unto you; But if ye show ingratitude, truly My punishment is terrible indeed.”.

All praise is for Allah, the exalted, for his favours that cannot be counted.

I dedicate this work to my Parents, my brother, my sisters, my relatives, my friends, and for all those who helped, supported and taught me.

Table of Contents

List of Table

List of Figures

List of Abbreviations

AI	**Artificial Intelligence**
IDS	**Intelligent Database System**
BFS	**Breadth First Search**
DFS	**Depth First Search**
UI	**User Interface**

Design of Intelligent Database System for Vehicle Fault Diagnosis

Prepared by :

Saeed Mohammad Saeed Tayem

Supervised by :

Dr. Hussein H. Owaied

ABSTRACT

This thesis presents a design and implementation of Intelligent Database System for Vehicle Fault Diagnosis using mix of many knowledge representation forms.

The scheme for knowledge representation uses both procedural and declarative knowledge representation formalisms through the application of relational database. So the rule base, case base and frame base formats have been converted into tables. The scheme facilitates combination of forward and backward chaining reasoning, using the problem reduction method for solving problem, and the heuristic search technique. All the editing facilities of system; inserting, deleting and updating of a rule, case, and frame are present.

In this thesis, visual studio 2008 (VB.Net) have been used for the implementation of the system and suitable user interface design. The implementation is an application for the system in the domain vehicle fault diagnosis. Therefore, the Intelligent Database System for Vehicle Fault Diagnosis tested for many cases and the results, from different cases, matched the results of the vehicle mechanic.

تصميم قاعدة بيانات ذكية لتشخيص أعطال المركبات

إعداد

سعيد محمد سعيد تيم

إشراف

الدكتور حسين عويد

الخلاصة

تمثل هذه الدراسة تصميم وتنفيذ قاعدة بيانات ذكية في مجال تشخيص أعطال المركبات . باستخدام مزيج من أشكال تمثيل المعرفة .

المخطط الخاص بتمثيل المعرفة يستخدم أشكال تمثيل المعرفة الإجرائية والتعريفية من خلال تطبيق قاعدة بيانات علائقية .لذا فقد تم تحويل قاعدة الأحكام وقاعدة الحالات وقاعدة الأطر إلى جداول . مخطط يسهل الجمع بين المنطق تسلسل إلى الأمام والخلف، وذلك باستخدام أسلوب تجزئة المشكلة من أجل حلها ، وتقنية البحث الإرشادي. جميع أدوات تحرير النظام؛ الإدراج والحذف والتحديث تم استخدامها.

في هذه الدراسة تم استخدام برمجية فيجوال بيسك دوت نت 2008 لتنفيذ النظام وتصميم واجهة مستخدم مناسبة. كما تم تنفيذ هذا التطبيق في مجال تشخيص أعطال المركبة . نظام قاعدة البيانات الذكي اختبر في الكثير من الحالات وكانت النتائج، لمختلف الحالات متطابقة مع نتائج ميكانيكي السيارات.

Chapter One

INTRODUCTION

Chapter One

Introduction

1.1. Overview

Most of the artificial intelligence (AI) books state that there are many definitions of AI and this is due to the fact that authors came from different schools of thoughts and backgrounds. Owaied and Abu Arr'a (2007) defined Artificial Intelligence as "A.I. is a concept of study and research for finding relationship between cognitive science and computation theories in order to represent these relationships as either data structures, search techniques, problem solving methods or representation forms for knowledge and the final goal of AI is to build an intelligent machine". Nilsson (1971) gave an understanding and comprehensive overview for AI as an onion model shown in figure 1.1

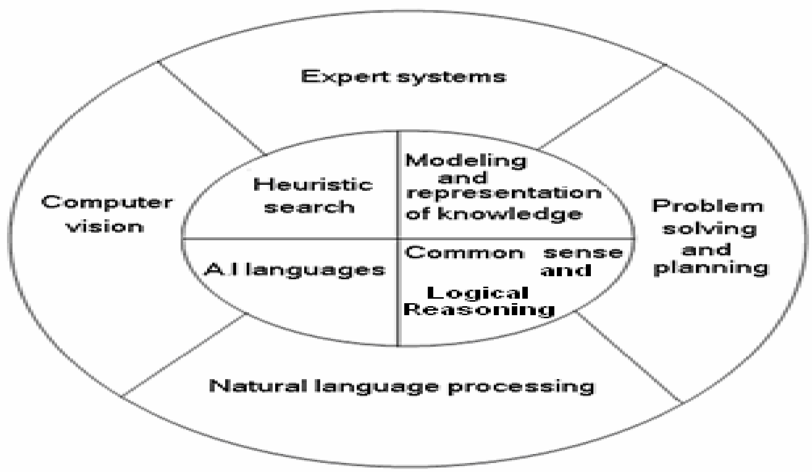


Figure 1.1: An overview of AI

The inner ring represents the tools of AI which are:

A. Modeling and representation of knowledge

B. Common sense and logic reasoning

C. AI Languages

D. Heuristic search

The outer ring represents the most important application areas of AI which are:

A. Expert systems

B. Problem solving and planning

C. Natural language processing

D. Computer vision

The Intelligent Databases (IDBs) are originated from the integration of databases technologies with artificial intelligence technologies. The IDBs are characterized by the presence of stored rules in a rule base and facts stored in a fact base. Altogether conforms to the knowledge base, in which different forms of reasoning are applied (Ana & Jose, 2007). The Intelligent Databases are developed for variety of domains like medicine, mathematics, engineering, chemistry, geology, computer science, business, law, defense and education. This thesis deals with types of problems related to diagnosis field and the implementation for the vehicles faults.

1.2. Problem Statement

Any system has the probability to breakdown anywhere and anytime, therefore there is a need for knowledge related with this system. The power of the intelligent database system depends on their knowledge base representation form. There are many schemes which have been used for knowledge representation, such as rule base, case base, model base, logic forms, and frame structure. Most of the earlier intelligent database systems concentrate on a specific type of problem and use one or two forms of knowledge representation. In addition to that, most of the existing intelligent database systems in the domain of diagnosis are built using one form only.

In this thesis the appropriate form will be of three types, these are rule base, case base and frame base. Therefore, there are many problems associated with the mixing of these three types of knowledge representation schemes in the design and implementation of intelligent database system. These problems are due to the difference of the methodology of the three forms and consequently affecting the design of:

1. The User Interface.
2. The problem solving method and search technique.

1.3. The Objectives of the proposed project

The main objectives of this research are:

1- Designing the proposed system for the diagnosis malfunction of vehicle as intelligent database system.
2- Mixing many knowledge representation forms of the proposed intelligent database system for diagnosis in order to facilitate the activities of human intellect forms, analogical reasoning, induction and deduction.
3- Applying the proposed intelligent database system for vehicle fault diagnosis.

1.4. Research Importance

The new enhancement in the Design of Intelligent Database System for Vehicle Fault Diagnosis can be applied in different environments as Medicine, Industry, Plants, Computer Science, Business, so as to save money and effort and reduce the time of work. So using three types of knowledge base representation forms; rule base, case base and frame base where most of the existing intelligent database system in the domain of diagnosis is built using one form only.

The goal for implementation of the proposed system in the field of vehicle diagnosis is to discover the reasons for malfunction, which may happen to the vehicle at anytime and anywhere and therefore call the knowledge stored in knowledge base.

1.5. Thesis Structure

The thesis includes six chapters; the current chapter is the introduction. Chapter two gives an overview of the knowledge engineering.

Chapter three is the literature survey for the thesis, showing the related work regarding intelligent database system.

In chapter four, we present the Design of Intelligent Database System for Vehicle Fault Diagnosis

In chapter five, we present the implementation of the proposed system in the field of vehicle diagnosis. Finally, chapter six contains the conclusion and the future work for the thesis.

Chapter Two

KNOWLEDGE ENGINEERING

Chapter Two

KNOWLEDGE ENGINEERING

2.1. Overview

Knowledge is a human familiarity about something that can include facts, information, descriptions, meta-knowledge and/or skills acquired through experience or education. Knowledge maybe classified in (Pezzulo, 2011):

1) Procedural knowledge is stored as a set of procedures which can themselves determine when they should be executed. Their execution is the intelligent behavior that was expected in the situation.

2) Declarative knowledge is stored as a set of statements about the world. These statements are static but can be added, deleted or modified.

2.2. Knowledge Engineering

Knowledge engineering is currently synonymous with modeling and representation of knowledge which is the most important tool of A.I. Knowledge engineering has the following processes (Russel and Norvig, 2003).

Identify the task: The range of questions that the knowledge base should be able to answer must be established. Further, the facts that should be included in the knowledge base have to be identified.

Assemble the relevant knowledge: The knowledge that should be included in the knowledge base is identified. This identification is done with regard to the task. Consequently, all the knowledge in the domain does not need to be included in the knowledge base if that is not required by the task.

Decide on a vocabulary: A vocabulary is chosen to represent the concepts of the domain. This vocabulary is known as the ontology of the domain. The ontology describes the existence of things but not their properties or interrelationships.

Encode general knowledge about the domain: General knowledge in the domain is included to the knowledge base. A good ontology will result in less general information in the knowledge base.

Knowledge engineering is the practical work of elicitation, collection, acquisition of knowledge, and proposing a suitable form for representing the knowledge (Jafari, Akhavan, Akhtari, 2011). This practical work can be done according to the following four phases; Knowledge acquisition and elicitation, Knowledge representation schemes, design the user interface and inference engine accordingly.

2.2.1. Knowledge Acquisition and Elicitation

The knowledge acquisition and elicitation is the first phase of knowledge engineering. There are many resources of knowledge such as: books, journals, internet, work fields, and humans. The most important resource is the human being. There are many methods used to elicit knowledge from human. These are:

1. **Interviewing**: Method used in all areas and the most common because it does not require prior knowledge of the domain, and has requirements, which are:
    a) Prior knowledge about the people to be interviewed (C.V).
    b) The interviewer should have experience and training of how to elicit knowledge from people.
    c) This method needs to interview more than one person.
    d) The time of the interview be limited
2. **Software packages**: this method is required to design a special software package. Its main task is knowledge elicitation. This software should be used friendly and easily.
3. **Observation**: This method requires watching the expert people, and has the following requirements:

a) Knowledge engineer should write down everything he notes during his observation.

b) Knowledge engineer should not interfere with the domain expert people in their work.

c) Take a long time because can't show all cases at the same time.

2.2.2. Knowledge Representation Schemes

Knowledge representation schemes can be defined as an adequately precise notation for representing knowledge. These schemes can be categorized as declarative representation schemes and procedural representation schemes (Sirikumara, 2006). There are many schemes which have been used for representation of knowledge, the following are descriptions of them.

2.2.2.1. Logic Representation Schemes

Logic is a very suitable tool for representing real world models. It can represent very complex relationships among objects, it can represent hierarchies, and it is very extensible (Gonzalez & Dankel, 1993) . There are many forms of logic, such as :

1. **Propositional Logic:** also called Boolean algebra and has been used in the design and implementation of all the integrated circuits as an expression of Boolean algebra of any integrated circuit.

2. **Predicate Logic:** There are two types of predicate logic. These are, first order predicate logic and multi-sorted predicate logic. The first order predicate logic is essential of the design of PROLOG language. The multi-sorted predicate logic is the bases of the design LISP programming language.

3. **Fuzzy logic:** will convert the quality into quantity in a process by computer system which is a digital data processing system (Zadeh, 1965).

2.2.2.2. Case Base Representation Scheme

This method is based on the knowledge of previous cases to draw conclusions about new cases. The known cases are stored in an explicit database and the new cases can be added to the database. Thereby this method is learnt with time. When it comes to solving a new problem the case-base reasoning method uses the following steps (Luger, 2005).

1. Retrieving suitable case from the database: Finding a similar case from the past situation is a difficult task because of the difficulty of defining the similarity between two cases. Organizing the storage of the cases and retrieval of cases are central for effective case-based reasoning method. There are several heuristics to cope with this. Cases can be organized by the goal and retrieved when the case has the same goal as the current situation. Another organizing method is to use cases with the most important features matched or the most number of features matched. The matching may first look for exactly matched cases before looking for a more general case. Using the cases most frequently matched or most recently matched is also used when retrieving cases to match a new situation. Another method is to use the case that matches without much adjusting. Using these heuristics a similar case is retrieved.

2. Adjust a retrieved case to the current situation: An already established case contains the operations for transforming the initial state to the goal state. The reasoning mechanism has to determine the operations to transform the established case to the current situation. If analytical methods are available, then they are used otherwise heuristic methods are applied to find out the operations for the new case.

3. Utilizing the adjusted case The modified case is tested against the current situation and adjusted until it successfully solves the current situation.

4. Saving the solution with information about the success This requires updating the indexing structure. This is beneficial for solving similar problems in the future. Further, the system is learnt by adding new cases this way.

2.2.2.3. Rule Base Representation Schemes

Sometimes just referred to as rules or productions. These are two part statements that embody small pieces of knowledge. The first part of the rule called the condition, expresses a situation or premise while the second part, called the action, states a particular action or conclusion that applies if the situation or premise is true. The first or left-hand part of the rule is a statement with the prefix IF. The second or right-hand part of the rule is a statement with the prefix THEN.

The rule base consists of three major components which are: Working Memory, Rule Base and the Interpreter (Antonsen & Viazzi , 2006)

1) The working memory is the place where all the processes of the system will be done and usually contains the following (Girratano, 1998):

1. facts about the world
2. Can be facts observed directly or derived from a rule
3. temporary knowledge about the problem-solving session
4. May be facts modified by application of the rules.

2) The Rule Base is a collection of rules, each rule is a step in a problem solving process. Rules are persistent knowledge about the domain.Typically, only modified from the outside of the system, e.g. by an expert in the domain. The syntax of rule is a IF <conditions> THEN <actions>

The conditions are matched to the working memory, and if they are fulfilled, the rule may be fired.

The Actions can be:

1. Adding fact(s) to the working memory.
2. Removing fact(s) from the working memory.
3. Modifying fact(s) in the working memory.
4. Driving new facts to the working memory

3) The interpreter controlling the application of the rules for a given assertions in the working memory. For each cycle of the Interpreter, the following actions will happen:

1. Selection: Select the rules from the rule base according to the goal statement.
2. Matching: Match the selected rules according the given assertions in the working memory.
3. Confliction: Choose the most economic rule, in term of minimum space and processes, from the matched rules.
4. Firing: Execute the chosen rule.

2.2.2.4. Frame Representation Schemes

Marvin Minsky (1974) proposed the concept of frames as structures where each frame has its own name and a set of attributes, or slots, associated with it. Luger (2005) described what is necessary to use frames and provide a natural way for the structure and representation of knowledge. In a single entity, a frame combines all necessary knowledge about a particular object or concept. The frame provides a means of organizing knowledge in slots to describe various attributes and characteristics of the object.

2.2.2.5. Model Base Representation Schemes

A model is a representation containing the essential structure of some object or event in the real world, The representation may take two major forms (Stockburger, 2007) :

1) Physical, as in a model airplane or architect's model of a building

2) Symbolic, as in a natural language, a computer program, or a set of mathematical equations.

Sirikumara (2006) said that “The key strength of this approach is the ability to represent the functional and structural knowledge of the domain in the knowledge base. Therefore, the systems based on this approach possess the ability to cope with new problems which were not taken into consideration during the initial designing process. This approach is considered to be robust, thorough and flexible because it falls back to first principles as a human would do in the case of new problem. The capability to provide with causal explanation is also another major advantage. This ability provides the users with greater understating of the problem situation".

2.2.3. Design User Interface

User interface is the method by which the system interacts with a user. These can be through dialog boxes, command prompts, forms, menu or other input methods. Some systems interact with other computer applications, and do not interact directly with a human. In these cases, the expert system will have an interaction mechanism for communication with the other application via interface port.

2.2.4. Design Inference Engine

Design and implementation of the inference engine depends on the representation of knowledge in the knowledge bases. The design and implementation of inference engine will be regarded as a combination of problem solving method, reasoning agent and search technique. The reasoning agent is responsible to accept sophisticated queries concerning some specific problem and to execute appropriate knowledge (Owaied and Abu Arr'a, 2007). There are two kinds of problem solving methodologies used in inference engine; forward chaining and backward chaining systems. In a forward chaining system, the initial facts are processed first, and keep using the rules to draw new conclusions given those facts. In a backward chaining system, the hypothesis (or solution/goal) we are trying to reach is processed first, and keep looking for rules that would allow concluding that hypothesis. As the processing progresses, new sub goals are also set for validation. Forward chaining systems are primarily data-driven, while backward chaining systems are goal-driven.

There are many techniques used to search such as depth first search, breadth first search, heuristic search, and so on. Usually the problem space represented as graph with the start point as the root of graph.

A graph consists of nodes and a set of arcs or links connecting pairs of nodes. Nodes are used to represent discrete states. Arcs are used to represent transitions between states. In expert systems states describe knowledge of a problem instance. The graph represented in figure 2.1. States are labeled (A, B, C ...). (Luger, 2005)

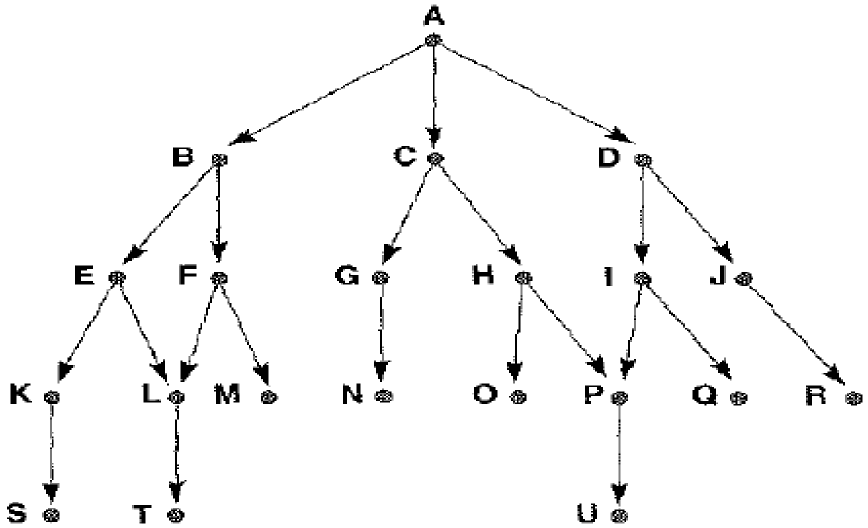


Figure 2.1: Graph for Breadth and Depth First Search

1) Depth First Search

In Depth-First Search (DFS), when a state is examined, all of its children and their descendants are examined before any of its siblings. DFS goes deeper into the search space whenever this is possible. Only when no further descendants of a state can be found arc its siblings considered. DFS examines the states in the graph of figure 2.1 in the order A, B, E, K, S, L, T, F, M, C, G. N, H, 0, P, U, D, I, Q, J, R. (Joyner, Nguyen & Cohen 2011).

Figure 2.2 present depth first search algorithm using list (Luger, 2005), on the graph of figure 2.1.

```
begin
   open := [Start];                                                    % initialize
   closed := [ ];
   while open ≠ [ ] do                                                 % states remain
      begin
         remove leftmost state from open, call it X;
         if X is a goal then return SUCCESS                            % goal found
            else begin
               generate children of X;
               put X on closed;
               discard children of X if already on open or closed;     % loop check
               put remaining children on left end of open              % stack
            end
      end;
   return FAIL                                                         % no states left
end.
```

Figure 2.2: Depth First Search Algorithm

2) Breadth First Search

Breadth-First Search (BFS) explores the space in a level-by-level fashion. Only when there are no more states to be explored at a given level does the algorithm move on to the next level. A breadth-first search of the graph of Figure 2.1 considers the states in the order A, B, C, D, E, F, G, H, I, J, K, L, M, N, 0, P, Q, R, S, T, U. (Joyner, Nguyen & Cohen 2011).

Figure 2.3 present BFS algorithm using list (Luger, 2005), on the graph of figure 2.1

```
function breadth_first_search;

begin
   open := [Start];                                                    % initialize
   closed := [ ];
   while open ≠ [ ] do                                                 % states remain
      begin
         remove leftmost state from open, call it X;
            if X is a goal then return SUCCESS                         % goal found
               else begin
                  generate children of X;
                  put X on closed;
                  discard children of X if already on open or closed;  % loop check
                  put remaining children on right end of open          % queue
               end
      end
   return FAIL                                                         % no states left
end.
```

Figure 2.3: Breadth First Search Algorithm

3) Heuristic Search

The heuristic search is a search technique most commonly used by humans in order to find the faster solution but not necessarily the optimal solution. This technique is based on heuristic information which is usually founded in the problem space. (Abbass, Sarker & Newton, 2002).

Chapter Three

LITERATURE SURVEY AND RELATED WORKS

Chapter Three

Literature Survey and Related Works

3.1. Overview

In recent years, intelligent database system attracted many researchers in the field of artificial intelligence; so the literature survey and related works of intelligent database systems will be presented in the following sections.

3.2. An Intelligent Database Application for the Semantic Web

El-Helw & Aly (2004) proposed techniques of the semantic web, to create formal definition rules for some kind of unstructured data, and to query this data for information that might not be explicitly stated, and that would otherwise be very difficult to extract. For this purpose, developed an integrated system that can extract data from both unstructured and structured documents, and then answer users' queries using the extracted data together with some inference rule that help to deduce more information from this data. As an example of this data, take the social relationships found in death-ads in newspapers. These ads contain both implicit and explicit information about various relationships among people. These relationships, if arranged and organized in a proper way, can be used to extract and infer other hidden, not explicitly mentioned relationships.

3.3. A Parallel Algorithm for Query in Deductive Databases

Arman (2006) presents an efficient algorithm to solve the generalized fully instantiated same generation query in deductive databases. The algorithm exhibits some intelligence by focusing on the relevant portion of the graph/database rather than considering all source nodes of the graph. The algorithm uses special data structures, namely, a matrix representation of the graph, representing the two-attribute normalized database relation, and a reverse matrix representation of the reverse graph. and also presents a performance study of the algorithm, and shows the advantages of the techniques used in the algorithm in solving the generalized form of the fully instantiated same generation query in deductive databases.

3.4. Query Processing in Intelligent Database Management System

Thuraisingham (2010) discussed techniques to secure query processing in various intelligent database systems. In particular, the following systems were considered

- relational systems
- systems based on distributed architecture
- fuzzy systems
- object oriented - semantic systems

3.5. Design of Intelligent Layer for Flexible Querying in Databases

Nihalani, Silakari, Motwani (2009) presented an innovative approach for the design of an intelligent database system for performing flexible queries in databases. An intelligent layer has been designed and incorporated into the existing database systems. The presented system accepts flexible user queries and converts them into a standard SQL query. Expression mapping, stop words removal, semantic matching and Levenshtein distance measure techniques have been utilized by the intelligent layer in the formation of the SQL query. The usefulness of the presented system has been demonstrated with the aid of experimental results.

3.6. A Frame-based Object-Relational Database System

Rattanaprateep and Chittayasothron (2006) present a frame-based system architecture that has an inference engine on both the client consulting expert system and on the knowledge base. Inferences that are performed on the client side are mainly user's interviews and interactive fact gathering. Inferences on the knowledge base side are performed based on already known facts recorded on the databases. Frames are implemented using object relational database technology.

3.7. Intelligent Databases Assist Valuation of Ecosystem Services

Villa, Ceroni, Krivov (2006) described the role of next-generation, intelligent databases (IDBs) in assisting the activity of valuation. Such databases employ artificial intelligence to inform the transfer of values across contexts, enforcing comparability of values and allowing users to generate custom valuation portfolios that synthesize previous studies and provide aggregated value estimates to use as a base for secondary valuation. Also introduced the Ecosystem Services Database, the first IDB for environmental valuation to be made available to the public, describe its functionalities and the lessons learned from its usage, and outline the remaining needs and expected future developments in the field.

3.8. Intelligent Database Management

Touch (1994) presented architecture for the data management software system of the intelligent database management of the distributed active archive center developed by stepwise refinement, and discussed how existing protocols are sufficient for use in this architecture to support both data ingestion and data fusion and visualization. The data management software system architecture presented is scalable, partitions the data management software system via gateway access servers, and include internally replicated processing components. A design in which control is distinct from data streams, both logically and topologically, has been shown, the architecture shown permits various implementations:

- Gateway as authenticator only remainder as centralized server.
- Gateway as delegator using vector pipelined remote evaluation (REV) processors.
- Gateway as authenticator using REV on workstations.

3.9. A Fuzzy System Design for Diagnosis of Prostate Cancer

Saritas, Allahverdi and Sert (2003) designed a fuzzy system for diagnosing, analyzing and learning purposes of the prostate cancer diseases to determine if there is a need for the biopsy. It also the user a range of the risk of cancer diseases. For this process it used prostate specific antigen. It was observed that this system was rapid, economical, without risk like traditional diagnostic systems, has also a high reliability and can be used as a learning system for medicine students. For the design process, prostate specific antigen, age and prostate volume are used as input parameters and prostate cancer risk is used as output. For fuzzification of these factors, the linguistic variables very small (VS), small (S), middle (M), high (H), very high (VH), very low (VL) and low (L) were used. Fuzzy Expert System: Parts of the developed fuzzy rules total of 80 rules are formed. This system is good for testing and learning process for the students, specializing in the prostate cancer diseases. The system does not answer if there is a cancer disease in the patient, but it gives a percentage of the possibility of the prostate cancer and helps the doctor to decide a biopsy or not.

3.10. An Application System for Hydro Electric Generator

Potter and Negnevitsky (2003) implemented an application using expert system called Level5 Object for scheduling hydro electric power in Tasmania. The system used frame base and rule base. It provides advice on distributing a power requirement across a hydro electric power system. This will be very useful for meeting the new demands placed on Hydro Tasmania by integration to the national power grid.

The cost of running modern power systems is very high and thus any improvement in the system operation can bring significant benefits. However, the power system operation is a difficult task as it includes such aspects as security, reliability, quality of power supply and the cost of running the system. Thus, the global objective of cost minimization has to be reached under specific technical constraints of the power system. The proposed application specifically aimed at scheduling and optimizing a hydro electric system. The proposed application implemented is to solve the problem of integration of a hydro electric system into a larger power grid, expressly Australia's national power grid. This integration will mean that Hydro Tasmania has to be able to schedule systems, according to demand, every five minutes.

3.11. A Fuzzy Frame Based System

Sharma, Kumar, Mustafa and Kumar (2003) proposed a novel approach, based on fuzzy logic and frames, for the creation of fuzzy expert shell. The fuzzy logic effectively deals with the type of communication normally used by humans, whereas the frames have been used to store the correlated information in the form of a block. This shell uses a local inference mechanism within the frame with the help of a multi criteria decision matrix. The User interface (UI) allows the user to communicate with the system. The user submits the query and additional information and when required by the system, through UI. It supplies this information to Query frame Construction Block for further processing. At the end, UI also supplies the final output to the user. The major advantage of this shell is that it can extract relevant information from the user, manage uncertain information and produce a credible advice to the user in an uncertain environment.

3.12. A Case Based System for Heart Diseases Diagnosis

Salem, Roushdy and Hodhod (2005) presented case-based medical system prototype that supports diagnosis of four heart diseases namely; mitral stenosis, left-sided heart failure, stable angina pectoris and essential hypertension. 110 cases were collected for 4 heart diseases. Each case contains 207 attributes concerning both demographic and clinical data. After removing the duplicated cases, the system has trained set of 42 cases for Egyptian cardiac patients. Statistical analysis has been done to determine the importance values of the case features. The system used two different techniques for the retrieval process and investigated namely; induction and nearest-neighbor approaches. The results indicate that the nearest neighbor is better than the induction strategy, where the retrieval accuracy were 100% and 53.8% respectively. Cardiologists have evaluated the overall system performance where the system was able to give a correct diagnosis for thirteen new cases.

3.13. JavaDON: An Open-Source System

Tomic´, Jovanovic and Devedzˇic (2006) described an open-source expert system shell based on the OBOA framework for developing intelligent systems called JavaDON, The central idea of this project was to make an easy-to-use and easy to-extend tool for building practical expert systems. Since JavaDON is rooted in a sound theoretical framework, it is well-suited for building even complex expert system applications. JavaDON knowledge representation scheme used rules and frames.

Another important feature of JavaDON is its capability of saving knowledge bases in XML format, thus making them potentially easy to interoperate with other knowledge bases on the Internet. So far, JavaDON has been used to build several practical expert systems, as well as a practical knowledge engineering tool to support both introductory and advanced university courses on expert systems. JavaDON facilitates introducing the process of building ES to inexperienced users, due to its highly intuitive graphical user interface; it allows the users to specify the content of the knowledge base they wish to create without the need to cover a specific knowledge representation language first. The expert systems developed with JavaDON can be either desktop or Web applications.

3.14. A Web Based System

Jain B., Jain A. and Srinivas (2008) presented the design and development of a web based system for Fault Diagnosis and Control of Power System Equipment and its role in developing an Intelligent Fault Diagnosis and Control Paradigm (IFDCP) package for power system equipment. A brief description of expert system architecture and issues involved in developing a web based expert system shell and the technology used are discussed. The concept of designing a web based expert system with a user friendly GUI is also discussed. The application of the shell to develop the package IFDCP for fault diagnosis and control of general power system equipment which provides online help for diagnosing faults of electrical power equipment and clearing them is also tackled .

3.15. System for Diagnosis of Diseases in Rice Plant

Sarma, Singh A. and Singh k. (2010) present an architectural framework of the system in the area of agriculture and describe the design and development of the rule based expert system, using the shell ESTA (Expert System for Text Animation). The designed system is intended for the diagnosis of common diseases occurring in the rice plant. An Expert System is a computer program normally composed of a knowledge base, inference engine and user-interface. The proposed expert system facilitates different components including decision support module with interactive user interfaces for diagnosis on the basis of response(s) of the user made against the queries related to particular disease symptoms. ESTA programming is based on logic programming approach. The system integrates a structured knowledge base that contains knowledge about symptoms and remedies of diseases in the rice plant appearing during their life span. The system has been tested with domain dataset, and results given by the system have been validated with domain experts.

3.16. Rule Based System for Software

AL Ahmar (2010) presented the modeling and development of a rule based system for selecting a suitable software development methodology according to software project features. By combining rule based knowledge representation with object oriented database modeling, the user interaction with the system is based on a user-friendly graphical interface.

Chapter Four

Design of Intelligent Database System for Vehicle Fault Diagnosis

Chapter Four

Design of Intelligent Database System for Vehicle Fault Diagnosis

4.1. Overview

The brain was the first processor humans used from the beginning to solve their problems, through creating new ideas or imitating the ways nature or animals used to live. The mid-twentieth century witnessed the invention of the computer, to form a turning point for humans life and the revolution of information. This invention opened the way for the scientists to allow machinists to mimic the actions and thinking of human beings themselves and thereby create a new science known as artificial intelligence.

Owaied, Abu-A'ra & Farhan (2010) said that "Most people know the term artificial intelligence concerning about how to build an intelligent machine. This machine should have certain capabilities such as: behaves like a human being, smart, problem solver of unstructured and complex problems as human does, understands languages, learner, and able to reason and analyze data and information, and so on". Knowledge-based system is an artificial intelligence application that uses the knowledge about a specific and narrow domain. The structure of knowledge based system depends on the proposed functional model of human system, which was constructed according to the direction of arrow in the left of figure 4.1 from top to bottom (Owaied H.H. and Abu Arr'a M. M., 2007).

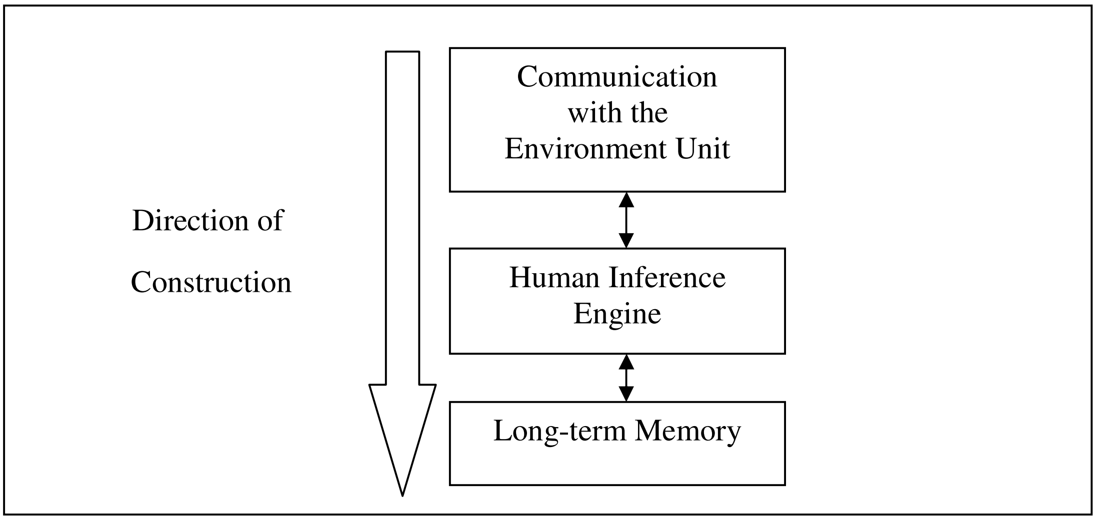


Figure 4.1: Functional Model of Human system

While the design and implementation of knowledge based system will be according to the direction of arrow in the left of figure 4.2 from bottom to top as mimic the human functional model. Therefore, the implementation starts from the knowledge base and then proposing an inference engine and a user interface which are suitable to the knowledge base representation forms.

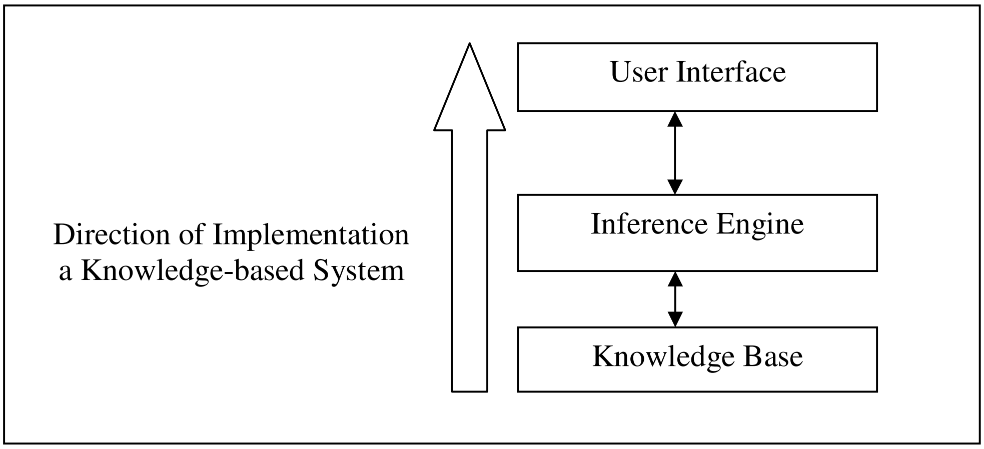


Figure 4.2: Structure of Knowledge-Based System

The most important phase in building knowledge based system is building the knowledge base. The implementation of knowledge base depends on the representation forms of the knowledge and usually there are many forms (Rule base, Case base, Frame base, Semantic nets, Logic forms and so on) used by human which may be applied.

Parsaye, Chignell, Khoshafian & and Wong (1989) defined intelligent databases as "databases that manage information in a natural way, making that information easy to store, access and use."

Ralston & Reilly (1993) defines IDB as " a database that contains knowledge about the content of their data. A set of validation criteria is stored with each data, for example maximum and minimum values or a list of the possible input".

The intelligent databases have as general purpose the generated and the discovery of information and knowledge. Among these types of databases we include the active, deductive, knowledge and fuzzy databases. In general the IDB are the natural evolution of the traditional databases, not only because they allow the manipulation of the data, also of the cognitive elements in form of facts and rules. One essential aspect of these databases is the possibilities of using techniques to discover knowledge, such as data mining techniques; all this permits learning patterns and data analysis strategies, as well as making classification and recognition, among others. The IDB systems are characterized by using an artificial intelligent technique that supports different reasoning mechanisms, they have a similar architecture to the expert systems that consist of a fact base, a rule base and must have persistence of the fact base (Ana & Jose, 2007).

4.2. Design of The Proposed Intelligent Database System

Figure 4.3 presents the Proposed Intelligent Database as Knowledge-based System. The proposed model consists of four modules, which are; user interface, inference engine, knowledge base, and editing facilities for knowledge bases. Most of the existing systems use one or two knowledge representation forms, the proposed system which uses three types of knowledge representation forms.

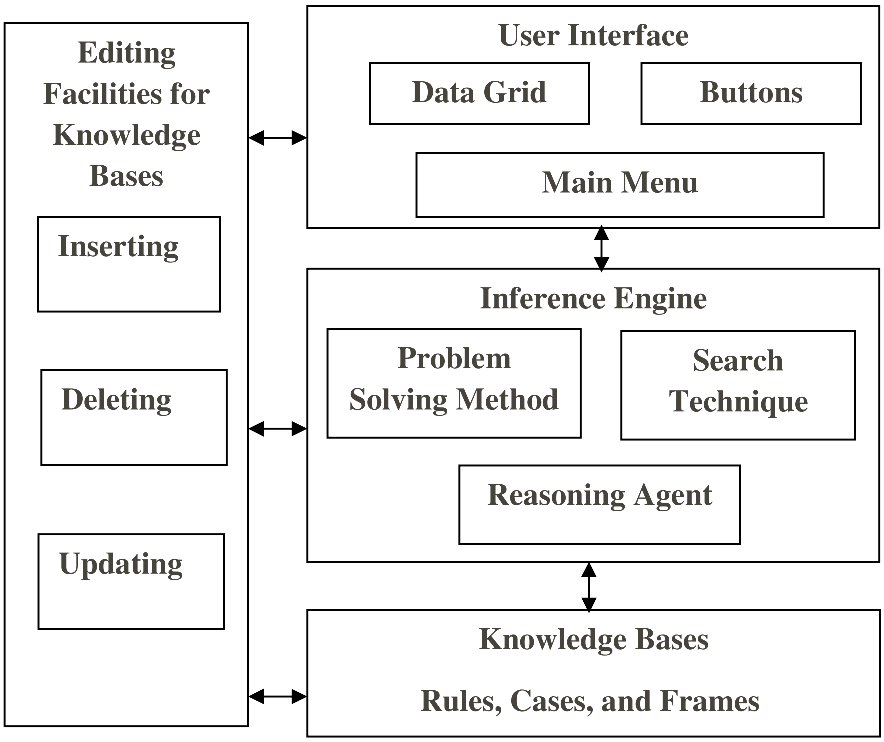


Figure 4.3: Architecture of the Proposed Intelligent Database System

The following subsections are detailed descriptions of the Design of Intelligent Database System for Vehicle Fault Diagnosis, a new hybrid scheme of knowledge representation using relational database as integrating of three different knowledge representation formats.

4.2.1. User Interface

The user interface simulates the communications with the environment unit of the functional model of human system. The communication between the user and the system is simplified by providing most of the facilities for the user to interact with the system. (Owaied H.H. and Abu Arr'a M. M., 2007). The user interface consists of three components:

- Main Menu: consists of several buttons. Each button represents a form; when the system starts this menu it will be displayed in order to allow the user to select one of the forms.

- Data Grid: A grid view or a data grid is a graphical user interface element that presents a tabular view of data. A typical grid view also supports the following:

  * Dragging column headers to change their size and their order.

  * In-place editing of viewed data.

  * Row and column separation, and alternating row background colors.

- Buttons: They are the controls which we click on to perform some action. Buttons are used mostly for handling events in code

4.2.2. Inference Engine

Implementation of the Inference Engine depends on the representation of knowledge in the knowledge bases of the proposed system. The implementation of inference engine will be regarded as a combination of problem solving method, reasoning agent and search technique. The reasoning agent is responsible to accept sophisticated queries concerning some specific problems to execute appropriate knowledge. The use of case base format will facilitate the analogical reasoning, the use of frame base format will facilitate the induction, the use of rule base format will facilitate the deduction. So the inference engine uses combination of forward and backward chains reasoning according to problem reduction method for solving problem, and the heuristic search technique.

4.2.3. Knowledge Bases

This thesis uses procedural and declarative knowledge representation formalisms. So the Rule, Case and Frame bases formats are used and converted into database tables through the application of relational data base. In the following subsections are descriptions of the three formats.

4.2.3.1 Case Base

Case base is a technique to solve problems by searching for a similar case from previous experience and then adapted to solve the problem. The Case-base has the following activities (Reisbeck & Schank, 1989) :

· Retrieve the most similar case or cases.

· Reuse the knowledge in that case to solve the problem

· Revise the proposed solution.

· Retain the solution as part of the new case.

The proposed method to organize the cases will be in three tables; the first table consists of two columns: column one presents the case number and column two presents the case name. The second table consists of two columns: column one presents condition number and column two presents condition name. The third table consists of three columns which present case number, condition number and condition Priority.

4.2.3.2 Rule Base

The rule base is a set of rules and the syntax of a rule is

IF <conditions> THEN <actions> format and usually called clausal form. The general clausal form is (Coenen, 1998)

$A_1, A_2, A_3... A_n \longleftarrow C_1, C_2, C_3... C_m$

In this thesis the relational database will be used to represent the rule as table as seen in Table 4.1. The rules will be stored in a table format with the maximum number of column is k, for instance, k=5, then (Col_{-1}, Col_{-2} … Col_{-5}). The first column represents the left-hand-side of the rule, which is the conclusion of a rule usually called action (A) and from column-2 to column-4 are used to represent the conditions of the rule ($C_1, C_2… C_5$), and the last column is the same action so this rule will be as Horn clause presented as follows:

$A_1 \longleftarrow C_1, C_2, C_3, C_4, C_5$

$A_2 \longleftarrow C_1, C_2, C_3$

Col-$_1$	Col-$_2$	Col-$_3$	Col-$_4$	Col-$_5$
A	C_1	C_2	C_3	A
A	C_4	C_5		
A_2	C_1	C_2	C_3	

Table 4.1: Layout of a Rule in the Table

4.2.3.3 Frame Base

Frame base is a knowledge representation that uses frames, a notion originally introduced by Marvin Minsky in 1974, as their primary means to represent domain knowledge. A frame is a structure for representing a concept or situation, frame consists of slots which can be filled by value, or procedures for calculating values (Chen ,Wu & Takagi, 1991).

In this thesis the relational database will be used to represent the frame as table as shown in table 4.2

Slot $_1$	Slot $_2$	Slot$_3$	Slot $_4$	Slot $_5$

Table 4.2: Layout of Frame in the Table

4.2.4. Editing Facilities for Knowledge Bases

This component is used to manage the facilities: inserting, deleting and updating processes for knowledge base. All these facilities are applied according to the given request by the end user.

Chapter Five

Implementation of Intelligent Database System for Vehicle Fault Diagnosis

Chapter Five

Implementation of Intelligent Database System for Vehicle Fault Diagnosis

5.1. Overview

A large segment of the vehicle driving population is constituted by drivers who have little or even less information regarding the diagnosis of a vehicle. Vehicle fault identification is not easy for a driver because it needs a lot of knowledge for finding the fault. Therefore, they extremely depend on expert mechanic.

In this thesis, mix of knowledge representation formats and the knowledge about the vehicle diagnosis have been acquired from vehicle's mechanics. By using two methods to elicit knowledge from human, these are interviewing and observing. Using both methods for collection of knowledge related to vehicle systems and malfunctions that occur for vehicles and the reasons for the malfunctions. This knowledge is included in the Intelligent Database System for Vehicle Fault Diagnosis

5.2. Knowledge Base Schemes

Since the knowledge bases, Rule base, Case base, and Frame, are converted into tables and usually the Databases were built in relational database systems. Therefore, relational database systems have been used for the implementation of knowledge bases schemes for the Intelligent Database System for Vehicle Fault Diagnosis.

The implementation consists of twelve tables, and they are: case table, condition table, case and condition table, seven tables for frames, rules table, and condition –frame.

5.2.1. Case Base for Vehicle Diagnosis

The proposed Scheme to organize the cases will be in three tables which are Cases table, Conditions table and Case-Condition table. While the case is a malfunction and condition is a cause of the malfunction.

Cases table contains two columns, the first column labeled by Case-No, while the second columns labeled by Case-Name, a set of cases were stored in the Case-Name column as shown in figure 5.1. The Column Case-Number is assigned as primary key.

Conditions table contains two columns, the first column labeled by Condition- No while the second column labeled by Condition-Name, a set of conditions were stored in the Condition-Name column as shown in figure 5.2. The Column Condition-Number is assigned as primary key

Case-Condition table contains three columns, the first column labels by case number. The second column labeled by condition number and the third column labels by condition priority as shown in figure 5.3. The columns (case number, condition number) are Primary key. The column (case number) and the column (Condition number) are foreign key.

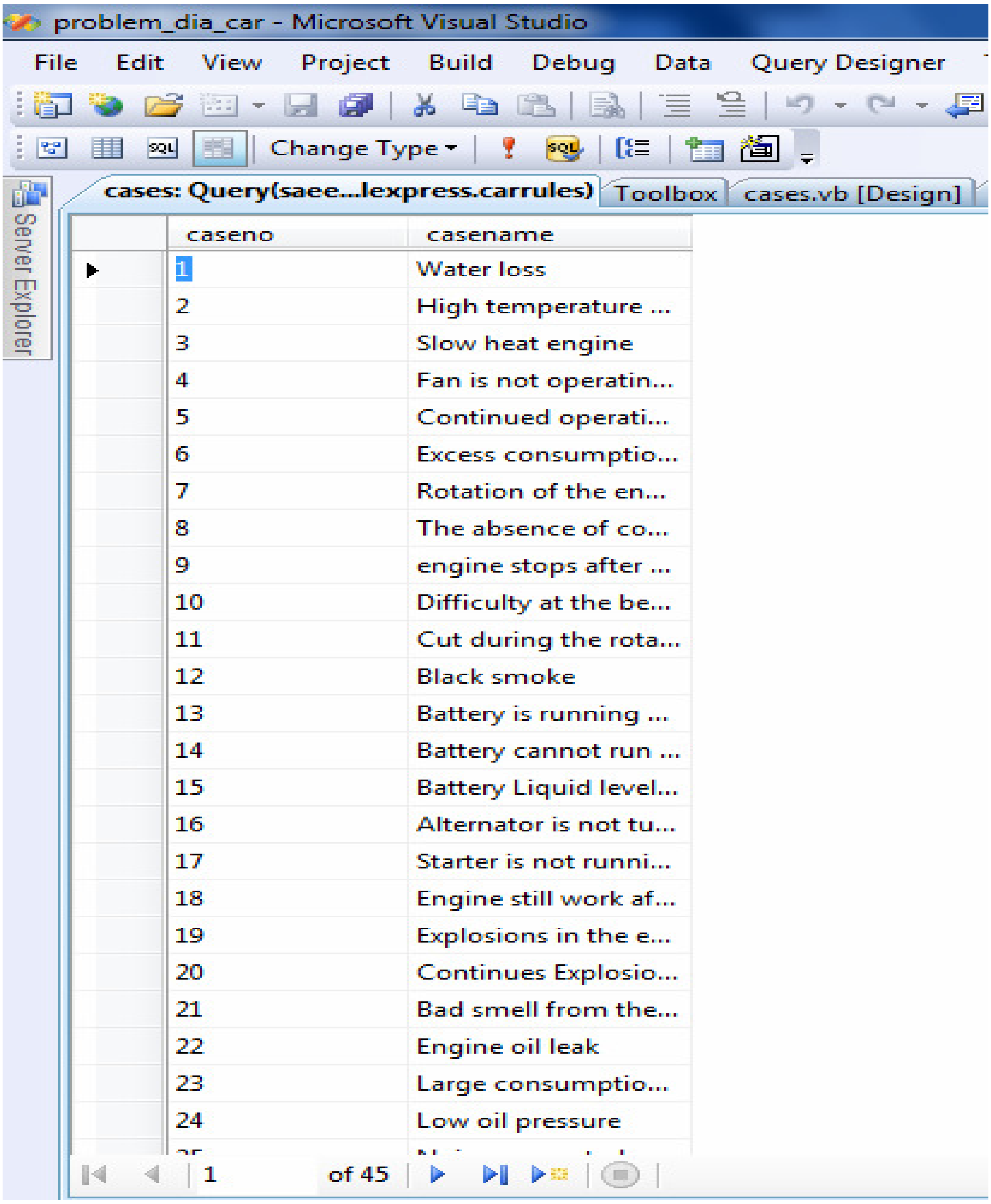
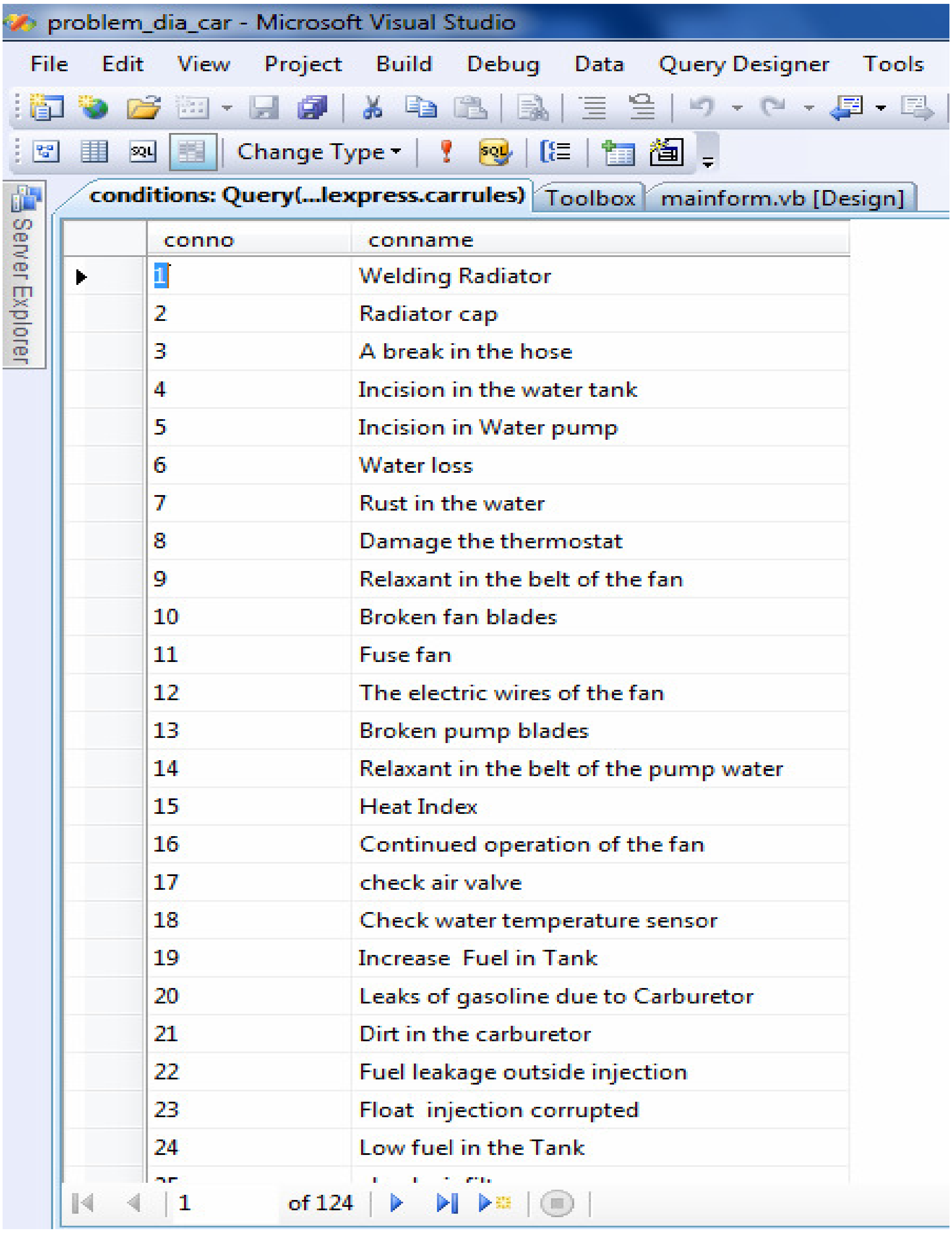
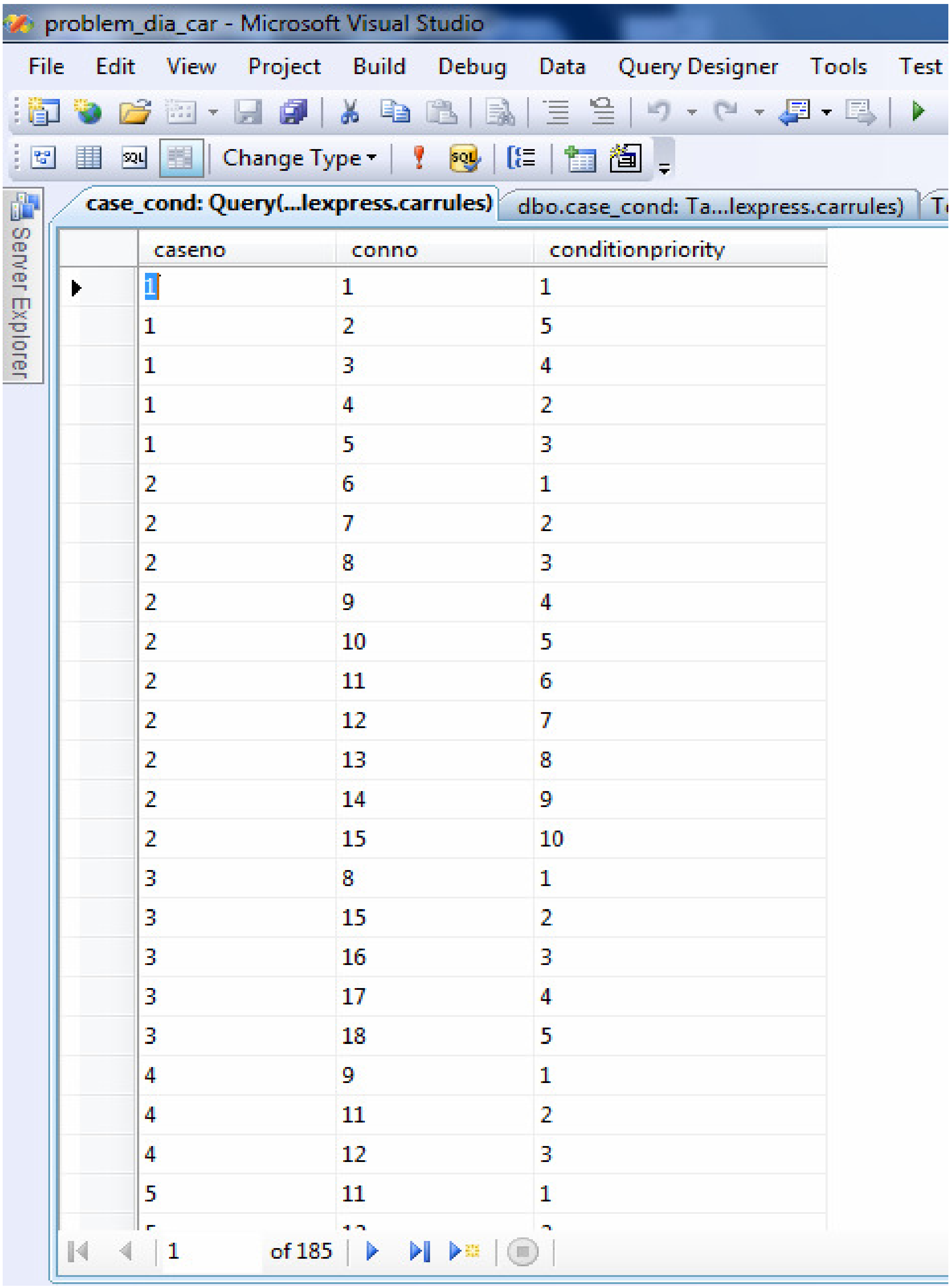


Figure 5.1: Cases Table



conno	conname
1	Welding Radiator
2	Radiator cap
3	A break in the hose
4	Incision in the water tank
5	Incision in Water pump
6	Water loss
7	Rust in the water
8	Damage the thermostat
9	Relaxant in the belt of the fan
10	Broken fan blades
11	Fuse fan
12	The electric wires of the fan
13	Broken pump blades
14	Relaxant in the belt of the pump water
15	Heat Index
16	Continued operation of the fan
17	check air valve
18	Check water temperature sensor
19	Increase Fuel in Tank
20	Leaks of gasoline due to Carburetor
21	Dirt in the carburetor
22	Fuel leakage outside injection
23	Float injection corrupted
24	Low fuel in the Tank

Figure 5.2: Conditions Table



caseno	conno	conditionpriority
1	1	1
1	2	5
1	3	4
1	4	2
1	5	3
2	6	1
2	7	2
2	8	3
2	9	4
2	10	5
2	11	6
2	12	7
2	13	8
2	14	9
2	15	10
3	8	1
3	15	2
3	16	3
3	17	4
3	18	5
4	9	1
4	11	2
4	12	3
5	11	1

Figure 5.3: Case-Conditions Table

Figure 5.4 presents the relationships between tables, table 5.1 and table 5.2, both present the key types used for tables and the relationships between tables respectively, table 5.3 presents the pseudo code used to create tables and relationship between tables.

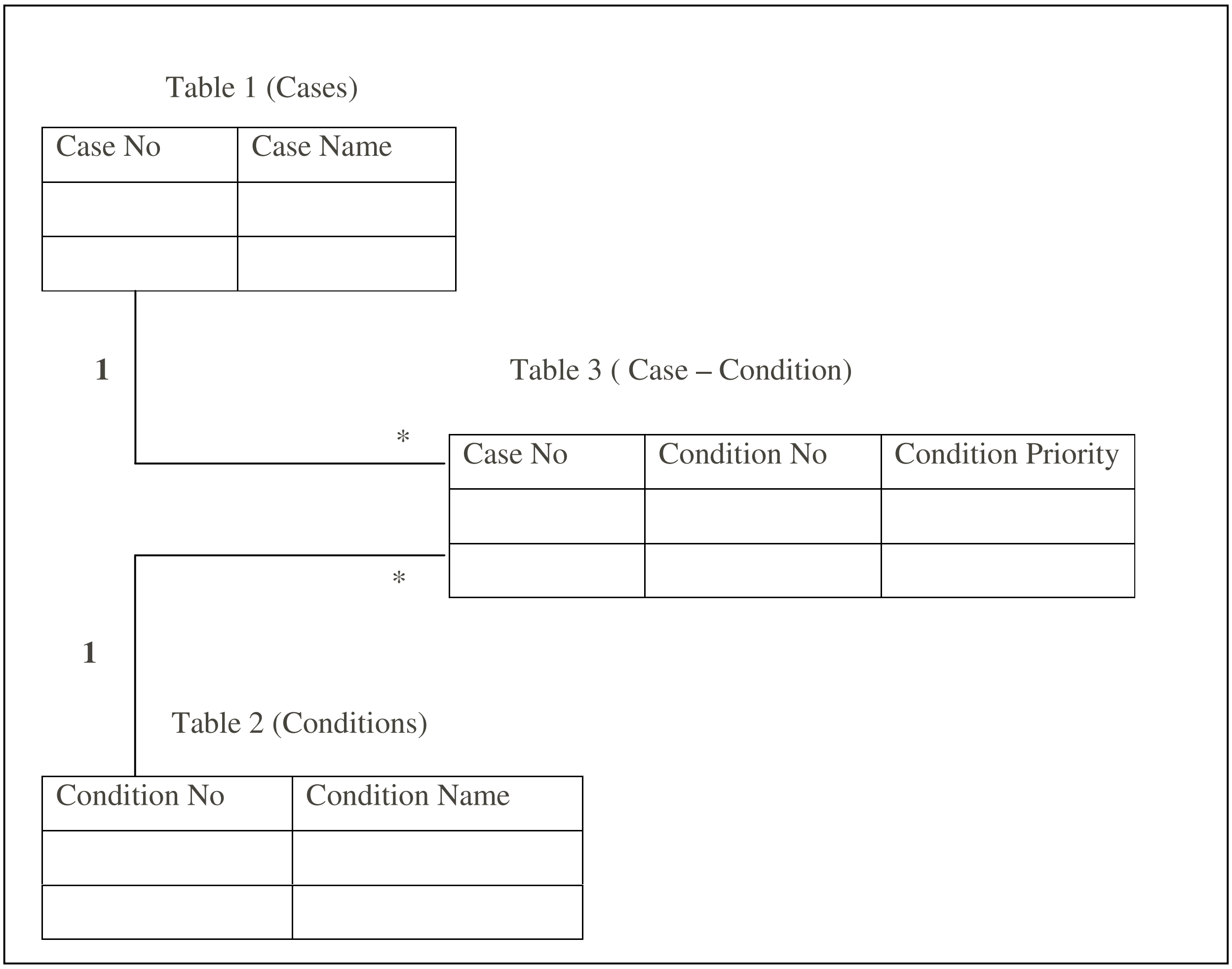


Figure 5.4: The Relationships between Tables

Table (1) The Column (Case_Number) is Primary key
Table (2) The Column (Condtion_Number) is Primary key
Table (3) The Columns (Case_Number, Condtion_Number) is Primary key
Table (3) The Column (Case_Number) is foreign key
Table (3) The Columns (Condtion_Number) is foreign key

Table 5.1: The Keys Types Used for Tables

Table (1) To Table (3)	One – To – Many
Table (2) To Table (3)	One – To – Many

Table 5.2: The Relationships between Tables

```
CREATE TABLE  [cases] (
          [caseno] [int] NOT NULL,
          [casename] [nvarchar] (250)  AS NULL,
 CONSTRAINT [PK_cases] PRIMARY KEY [caseno] )

 --------------------------------------------------------------------------------

CREATE TABLE   [conditions] (
          [conno] [int] NOT NULL,
          [conname] [nvarchar] (250)  AS NULL,
 CONSTRAINT [PK_conditions] PRIMARY KEY [conno] )

--------------------------------------------------------------------------------

CREATE TABLE  [case_cond] (
          [caseno] [int] NOT NULL,
          [conno] [int] NOT NULL,
          [casepriority] [int] NULL,
 CONSTRAINT [PK_case_cond] PRIMARY KEY [caseno], [conno])

ALTER TABLE [case_cond]  WITH CHECK ADD  CONSTRAINT [FK_case_cond_cases]
FOREIGN KEY ([caseno]) REFERENCES [cases] ([caseno])
ALTER TABLE [case_cond] CHECK CONSTRAINT [FK_case_cond_cases]

ALTER TABLE [case_cond]  WITH CHECK ADD  CONSTRAINT [FK_case_cond_conditions]
FOREIGN KEY([conno]) REFERENCES [conditions] ([conno])
ALTER TABLE [case_cond] CHECK CONSTRAINT [FK_case_cond_conditions]
```

Table 5.3: Present the Pseudo Code Used to Create Tables and Relationship Between Tables.

5.2.2. Rule Base for Vehicle Diagnosis

The rule base is a set of rules and the syntax of a rule is IF <conditions> THEN <actions>, while the action is a malfunction and condition is a cause of the malfunction.

$A_1, A_2, A_3 \ldots A_n \longleftarrow C_1, C_2, C_3 \ldots C_m$

Relational database will be used to represent the rule as table. The rules will be stored in a table format with the maximum number of column 5. The first column represents action (malfunction), and from column-$_2$ to column-$_4$ are used to represent the conditions (cause of the malfunction), while the last column is the same action .But if a rule has more than three conditions, the fifth column will be sub-action which has the reset of the conditions and so on.

The following is the procedure for representing a rule in a table .the number of rows required according to the number of conditions of the rule.

1) Applying the algorithm to calculate number of rows, n.

2) If n<=3, then, the representation in the table as shown in table 5.4.

Col-$_1$	Col-$_2$	Col-$_3$	Col-$_4$	Col-$_5$
A	C_1	C_2	C_3	C_4

Table 5.4: The Layout of the Rule with Conditions <= 3

3) If n>3, then, the representation in the table as shown in table 5.5.

$Col\text{-}_1$	$Col\text{-}_2$	$Col\text{-}_3$	$Col\text{-}_4$	$Col\text{-}_5$
A	C_1	C_2	C_3	A
A	C_4	C_5	C_6	A
A	C_7	C_8	C_9	A
	C_{13}	C_{14}	C_{15}	

Table 5.5: The Layout of the Rule with conditions > 3

Based on that will be applied to the vehicle diagnosis system

Example (1):

Fan is not operating at high $\longleftarrow$ temperature Relaxant in the belt of the fan,

Fuse fan, The electric wires of the fan

This example, shown that the number to conditions is three which are stored in columns from Col_{-2} to Col_{-4} and the Col_{-5} not used, and the action will be stored in Col_{-1} as shown in table 5.6.

Col_{-1}	Col_{-2}	Col_{-3}	Col_{-4}	Col_{-5}
Fan is not operating at high temperature	Relaxant in the belt of the fan	Fuse fan	The electric wires of the fan	

Table 5.6: The Layout of the Rule in Example (1)

Example (2):

Water loss $\leftarrow$ Welding Radiator, Radiator cap, A break in the hose, Incision in the water tank, Incision in Water pump

This example, shown that the number to conditions are five which are stored in columns from Col_{-2} to Col_{-4}, the action will be stored in Col_{-1}, and Col_{-5} pretend the sub action and store it in $Col\text{-}_5$ as shown in table 5.7.

Col_{-1}	Col_{-2}	Col_{-3}	Col_{-4}	Col_{-5}
Water loss	Welding Radiator	Radiator cap	A break in the hose	Water loss
Water loss	Incision in the water tank	Incision in Water pump		

Table 5.7: The Layout of the Rule in Example (2)

5.2.2.1. Creating Rule Base Table

Table 5.8 present the pseudo code used to create tables.

```
CREATE TABLE [rulestable](
        [casename] [nvarchar](250)  AS NOT NULL,
        [cond1] [nvarchar](250 AS NULL,
        [cond2] [nvarchar](250) AS NULL,
        [cond3] [nvarchar](250) AS NULL,
        [samecase] [nvarchar](250) AS NULL )
```

Table 5.8: the Pseudo Code Used to Create Rules Tables

5.2.2.2. Building Rules in Table

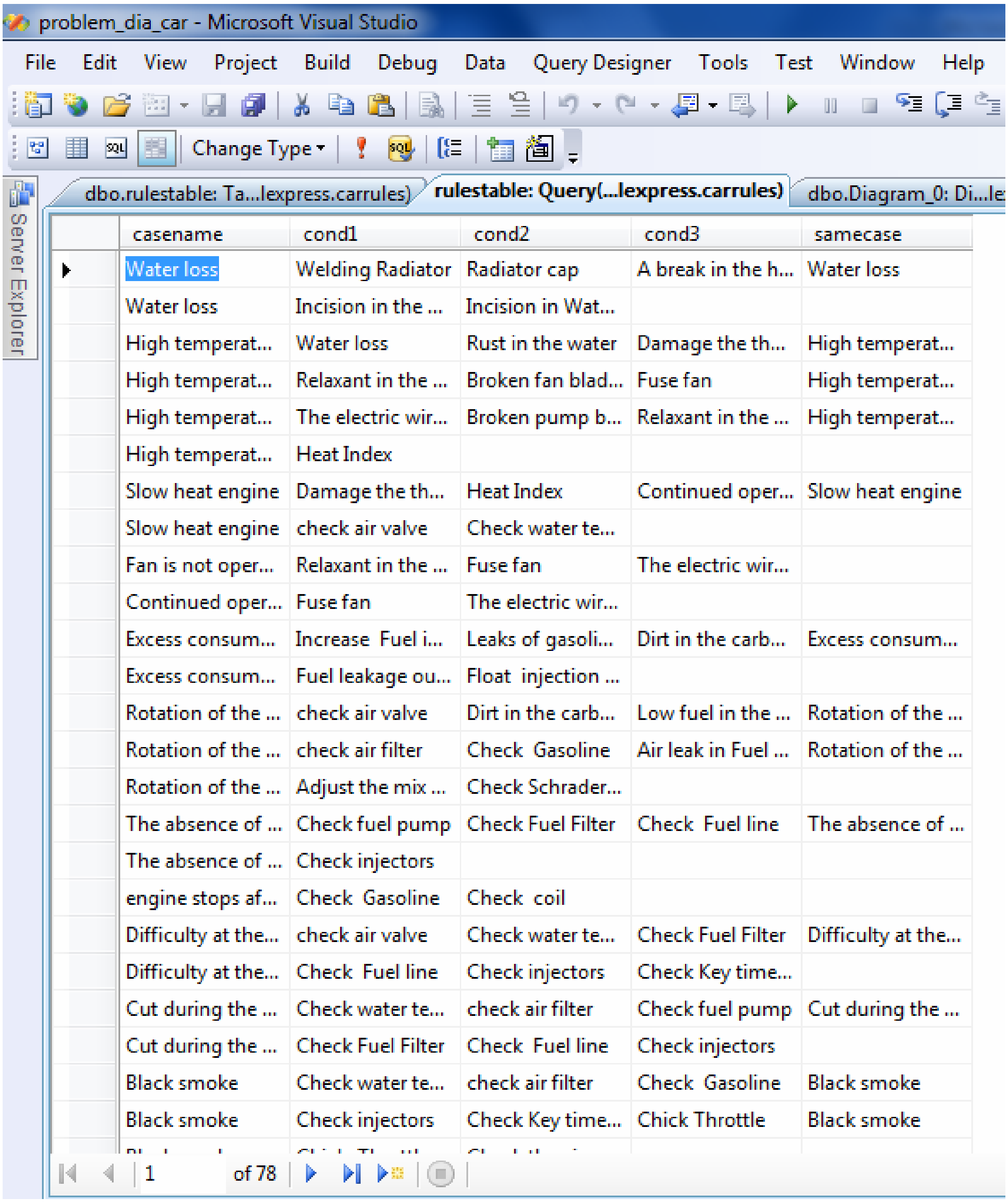
Figure 5.5 presents the way to build rules in a table; the table is built by using a table of cases and table of conditions. And figure 5.6 presents rules for the vehicle diagnosis system in a table.

```
Procedure Main()
Begin
Set Dataset1 equal to  (select distinct caseno   from   case_cond  order by caseno)
For loop from 1 to dataset1.count
Set Dataset2 equal to (select all rows in case_cond  where caseno equal caseno_item)
GOTO build_rules_table (dataset2)
Next Loop
End
--------------------------------------------------------------------------
Procedure build_rules_table (dataset2)
Begin
Create String array datarecord(5)
set mycount to dt.Rows.Count
If mycount is less or equal to 3 Then
Set Datarecord(1)= casename
```

```
For loop from 1 to dt.rows.count

Set Datarecord(loop)=(condition name)

Next loop

Connect to database

Insert_to_rules_table(datarecord(1), datarecord(2), datarecord(2), datarecord(4),
datarecord(5))


Else

set remainder to dt.Rows.Count Mod 3

If remainder equal 0 Then

Set Datarecord(1)=(case name)

Set Datarecord(5)=Null


For loop from 1 To dt.Rows.Count divide 3

Set Datarecord(loop)=(condition name)

If (recent loop) <> dt.Rows.Count divide 3 then

Set Datarecord(1)=(case name)

Set Datarecord(5)=(case name)

Else

Set Datarecord(1)=(case name)

Set Datarecord(5)= Null

Endif
```

```
Insert_to_rules_table(datarecord(1), datarecord(2), datarecord(3), datarecord(4),
datarecord(5))
Next loop
Else
For loop from 1 To dt.Rows.Count divide 3
Set Datarecord(loop)=(condition name)
Set Datarecord(1)=(case name)
Set Datarecord(5)=(case name)
Insert_to_rules_table(datarecord(1), datarecord(2), datarecord(3), datarecord(4),
datarecord(5))
Next Loop
For loop from dt.recent_index to dt.Rows.Count
 Set Datarecord(loop)=(condition name)
Set Datarecord(1)=(case name)
Set Datarecord(5)= Null
Next Loop
Endif
End
```

Figure 5.5: How to Build Rules in Table



casename	cond1	cond2	cond3	samecase
Water loss	Welding Radiator	Radiator cap	A break in the h...	Water loss
Water loss	Incision in the ...	Incision in Wat...		
High temperat...	Water loss	Rust in the water	Damage the th...	High temperat...
High temperat...	Relaxant in the ...	Broken fan blad...	Fuse fan	High temperat...
High temperat...	The electric wir...	Broken pump b...	Relaxant in the ...	High temperat...
High temperat...	Heat Index			
Slow heat engine	Damage the th...	Heat Index	Continued oper...	Slow heat engine
Slow heat engine	check air valve	Check water te...		
Fan is not oper...	Relaxant in the ...	Fuse fan	The electric wir...	
Continued oper...	Fuse fan	The electric wir...		
Excess consum...	Increase Fuel i...	Leaks of gasoli...	Dirt in the carb...	Excess consum...
Excess consum...	Fuel leakage ou...	Float injection ...		
Rotation of the ...	check air valve	Dirt in the carb...	Low fuel in the ...	Rotation of the ...
Rotation of the ...	check air filter	Check Gasoline	Air leak in Fuel ...	Rotation of the ...
Rotation of the ...	Adjust the mix ...	Check Schrader...		
The absence of ...	Check fuel pump	Check Fuel Filter	Check Fuel line	The absence of ...
The absence of ...	Check injectors			
engine stops af...	Check Gasoline	Check coil		
Difficulty at the...	check air valve	Check water te...	Check Fuel Filter	Difficulty at the...
Difficulty at the...	Check Fuel line	Check injectors	Check Key time...	
Cut during the ...	Check water te...	check air filter	Check fuel pump	Cut during the ...
Cut during the ...	Check Fuel Filter	Check Fuel line	Check injectors	
Black smoke	Check water te...	check air filter	Check Gasoline	Black smoke
Black smoke	Check injectors	Check Key time...	Chick Throttle	Black smoke

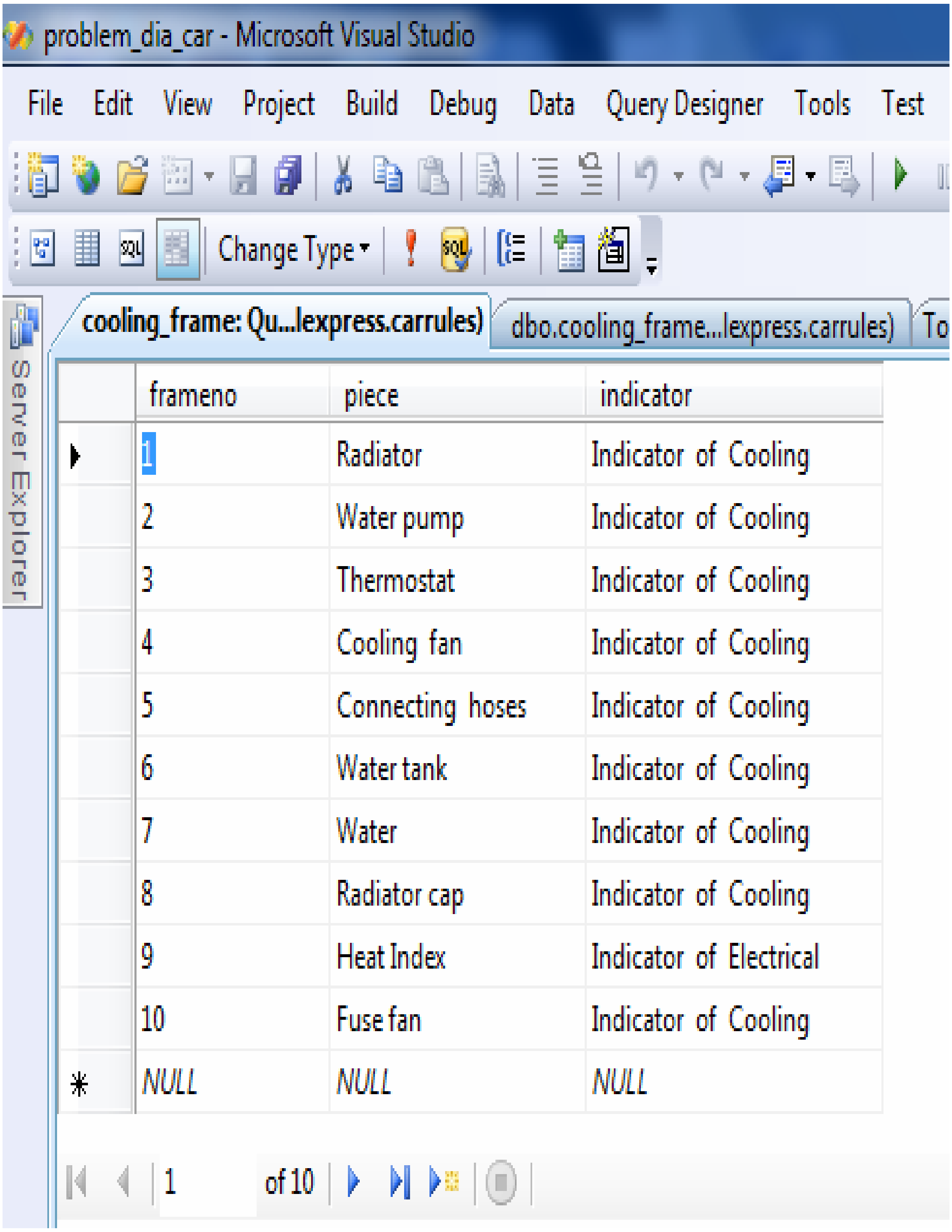
Figure 5.6: Rules for The Vehicle Diagnosis System in a table

5.2.3. Frame Base for Vehicle Diagnosis

In this Implementation we have seven frames which are: Cooling, Steering, Fuel, Electric, Engine, Gear Box, Brake. Each frame table contains three columns (slots), the first column is labeled by frame number. the second columns labeled by part (piece), and the third column labeled by indicator as shown in figure 5.7. The column frame number is the Primary key. Table 5.9 presents the pseudo code used to create frame tables.

```
CREATE TABLE  [frame_name]
        [frameno] [int] NOT NULL,
        [piece] [nvarchar](250) COLLATE Arabic_CI_AS NULL,
        [indicator] [nvarchar](250) COLLATE Arabic_CI_AS NULL,
 CONSTRAINT [PK_engine_frame] PRIMARY [frameno]
```

Table 5.9: Present the Pseudo Code Used to Create Frame Tables



frameno	piece	indicator
1	Radiator	Indicator of Cooling
2	Water pump	Indicator of Cooling
3	Thermostat	Indicator of Cooling
4	Cooling fan	Indicator of Cooling
5	Connecting hoses	Indicator of Cooling
6	Water tank	Indicator of Cooling
7	Water	Indicator of Cooling
8	Radiator cap	Indicator of Cooling
9	Heat Index	Indicator of Electrical
10	Fuse fan	Indicator of Cooling
NULL	NULL	NULL

Figure 5.7: Cooling Frame Table

- **Conditions and Frames Table**

This table contains three columns, the first column is labeled by condition number. the second column labeled by frame number, and the third column labeled by frame type as shown in Figure 5.8. The columns (condition number, frame number) are Primary key. The column (Condition number) is foreign key.

Frame type column represent frame table, if insert in frame number (one) this number means that the frame number follows the frame cooling.

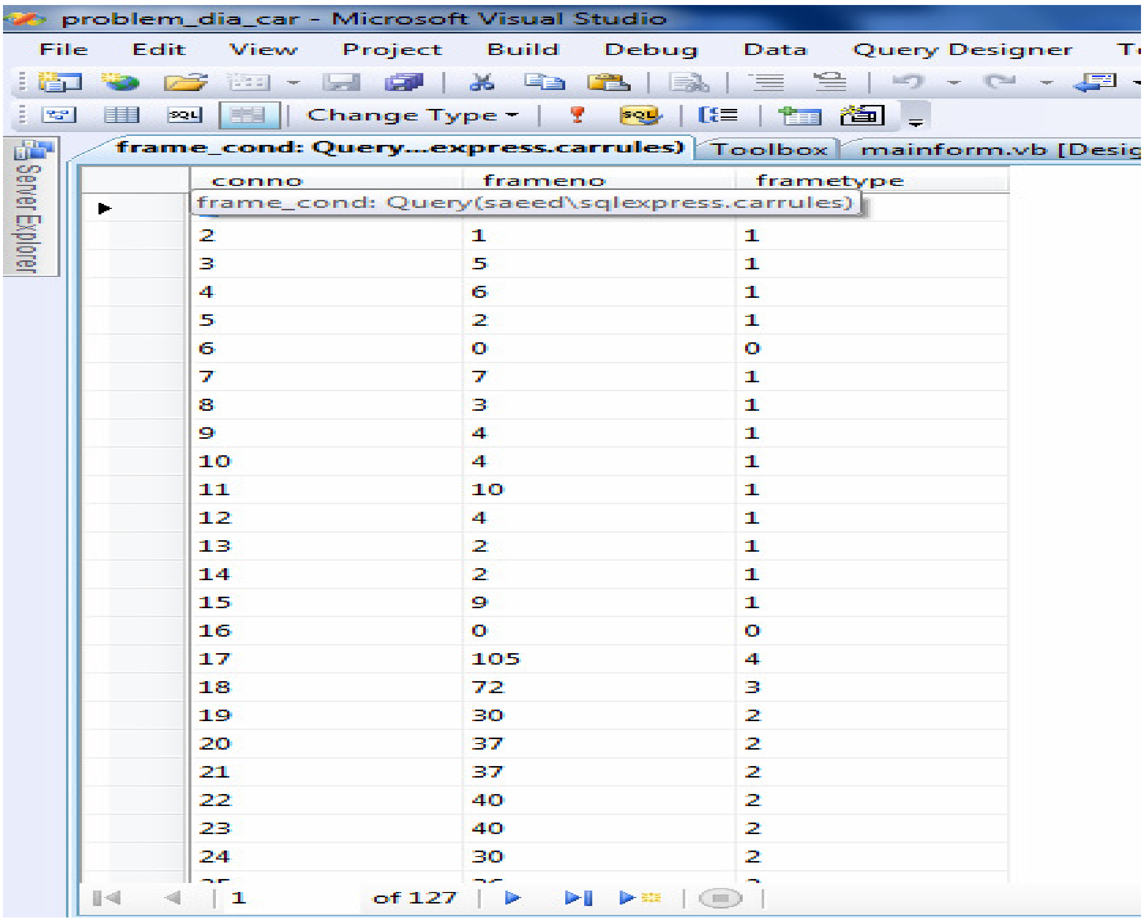


Figure 5.8: Condition and Frame Table

5.3. User Interface

The implementation of user interface consists of many visual studio 2008 (VB.Net) forms, such as the main menu, which consists of eight forms in two components (Editing Facilities, Explanation facilities) as shown in figure 5.9., when the system starts, this menu will be displayed in order to allow the user to select one of the following forms.

Figure 5.9: Main Menu

5.4. Editing Facilities for Vehicle Diagnosis System

This component is used to manage the facilities: inserting, deleting and updating process for knowledge base. Editing Facilities consists of four forms which are edit cases, edit conditions, edit frames, edit rules table as shown in figure 5.10.

Figure 5.10: Editing Facilities

5.4.1. Edit Cases

This form is used to edit the cases, which means the user can insert a new case to be added to the case base knowledge, delete an existing case, or update existing case as shown in figure 5.11.

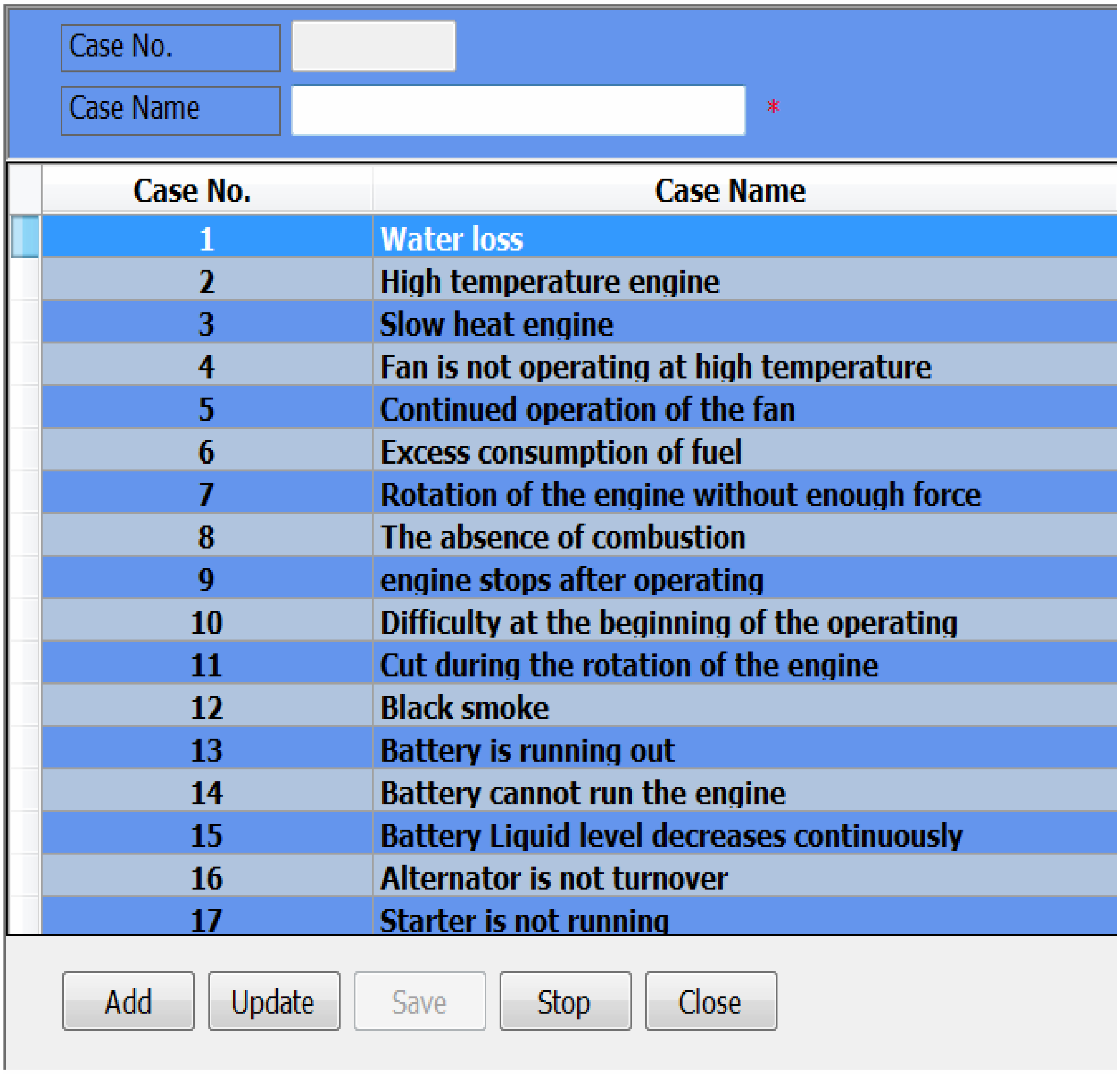


Figure 5.11: Editing Cases

5.4.2. Edit Conditions

This form is used to edit the condition, which means the user can insert a condition to be added to the rule base knowledge, delete one, or update existing condition as shown in figure 5.12, then the user has a capability of using editing facilities (insert, delete, and update) to the rule base knowledge.

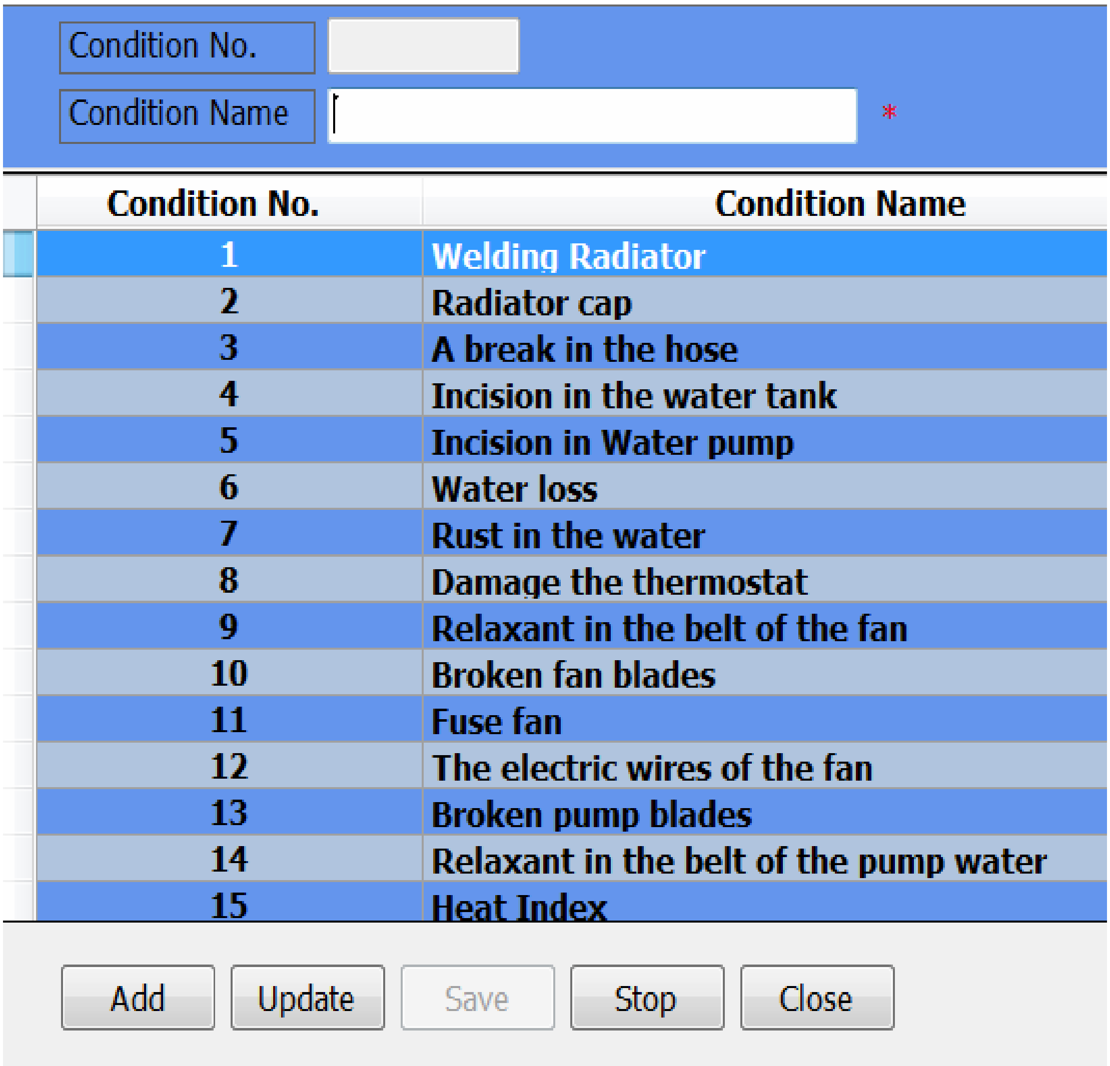


Figure 5.12: Editing Conditions

5.4.3. Edit Frames

This form is used to edit the frame, which means the user can insert a new piece to be added to the frame base knowledge, delete an existing piece, or update existing piece as shown in figure 5.13, then the user has a capability of using editing facilities (insert, delete, and update) to the frame base knowledge.

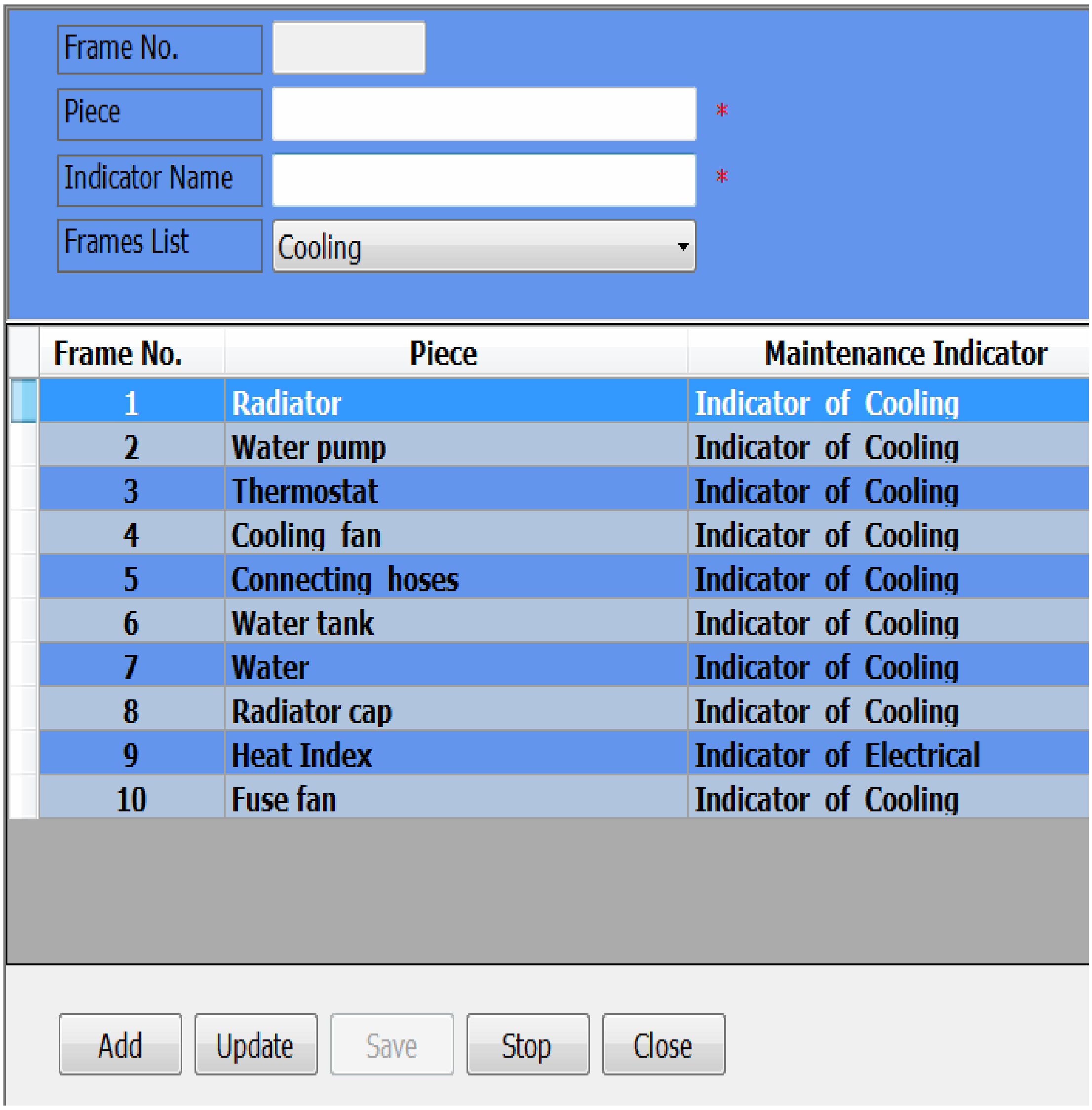


Figure 5.13: Editing Frames

5.4.4. Edit Rules Table

Figure 5.14 presents a table of rules, the table was built by using two other tables; the cases table and conditions table. The user can modify a rule by double clicking one of the rows.

Case Name	Condition 1	Condition 2	Condition 3	Case Name
Water loss	Welding Radiator	Radiator cap	A break in the hose	Water loss
Water loss	Incision in the water tank	Incision in Water pump		
High temperature engine	Water loss	Rust in the water	Damage the thermostat	High temperature engine
High temperature engine	Relaxant in the belt of the fan	Broken fan blades	Fuse fan	High temperature engine
High temperature engine	The electric wires of the fan	Broken pump blades	Relaxant in the belt of the pump water	High temperature engine
High temperature engine	Heat Index			
Slow heat engine	Damage the thermostat	Heat Index	Continued operation of the fan	Slow heat engine
Slow heat engine	check air valve	Check water temperature sensor		
Fan is not operating at high temperature	Relaxant in the belt of the fan	Fuse fan	The electric wires of the fan	
Continued operation of the fan	Fuse fan	The electric wires of the fan		
Excess consumption of fuel	Increase Fuel in Tank	Leaks of gasoline due to Carburetor	Dirt in the carburetor	Excess consumption of fuel
Excess consumption of fuel	Fuel leakage outside injection	Float injection corrupted		
Rotation of the engine without enough force	check air valve	Dirt in the carburetor	Low fuel in the Tank	Rotation of the engine without enough force
Rotation of the engine without enough force	check air filter	Check Gasoline	Air leak in Fuel feed line	Rotation of the engine without enough force
Rotation of the engine without enough force	Adjust the mix of air and Gasoline is not good	Check Schrader valve		
The absence of combustion	Check fuel pump	Check Fuel Filter	Check Fuel line	The absence of combustion
The absence of combustion	Check injectors			
engine stops after operating	Check Gasoline	Check coil		
Difficulty at the beginning of the operating	check air valve	Check water temperature sensor	Check Fuel Filter	Difficulty at the beginning of the operating
Difficulty at the beginning of the operating	Check Fuel line	Check injectors	Check Key time to run the injectors	
Cut during the rotation of the	Check water temperature	[illegible]	[illegible]	Cut during the rotation of the

Refresh Close

Figure 5.14: Editing Rules Table

5.5. Explanation facilities

Explanation facilities: Through conditions, cases and frames stored, the user can discover new facts and rules that make him able to get and take correct deduction. Editing Facilities consist of four forms which are Link Cases and Conditions, Link Conditions and Frames, Show Cases - Conditions and Frames, and Find Cases. as shown in figure 5.15

Figure 5.15: Explanation facilities

5.5.1. Link Cases and Conditions

Link Cases and Conditions form: the user can create a link between a case and conditions by entering the case's name in the field cases list. A list will be displayed it contains all cases that were stored before. After selecting the case, a set of conditions stored before will be displayed, by selecting and moving the conditions desired. The link process will be determined. And through the use of arrows priority conditions can be arranged. as shown in figure 5.16

The priority order of conditions (causes of the malfunction) can help us to correctly deduct and make the right decision to repair malfunction.

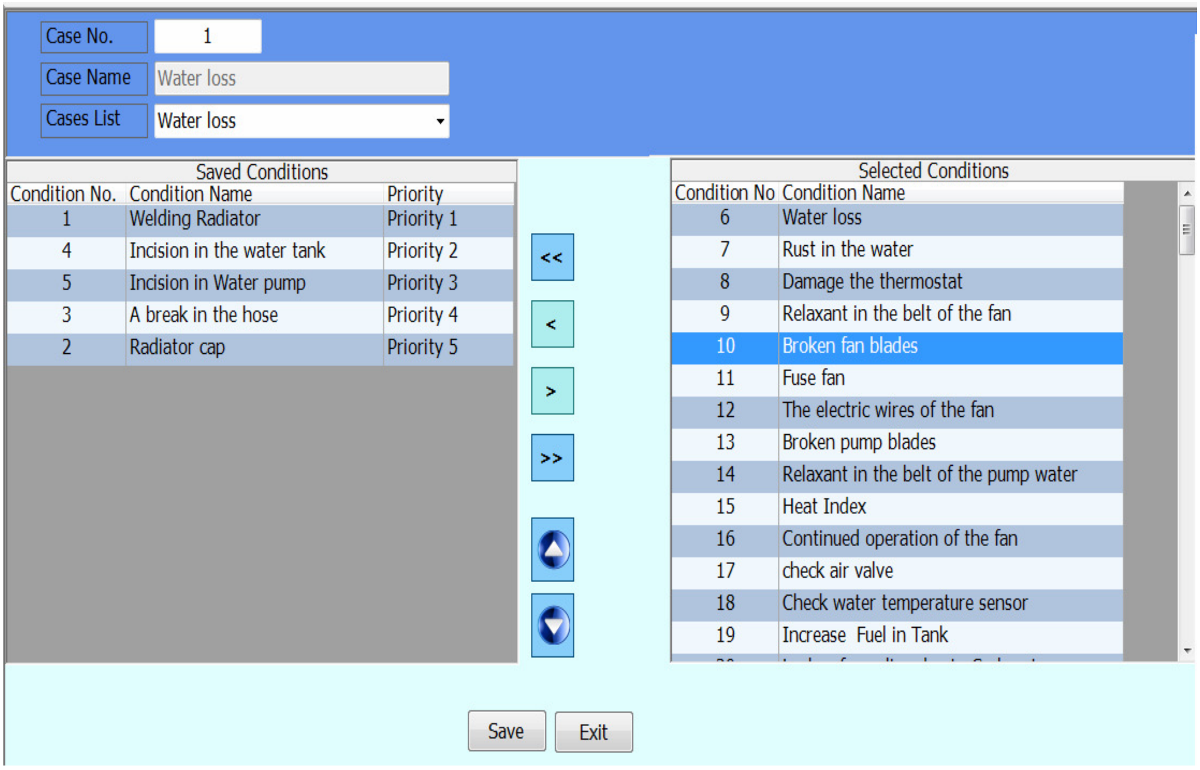


Figure: 5.16: Link Case and Condition

5.5.2. Link Conditions and Frames

In figure 5.17 the user can create a link between the conditions and frame by inserting the condition name in the field conditions list. A list will displayed it contains all conditions that were stored before. By choosing the frame's name off the field frame list, a list contains all frames stored before will be displayed. A list of pieces will be displayed when the user clicks on Execute button. The next step is selecting the part and clicking the save button the link process will be determined.

This linking process helps us to choose the right person to repair the failure.

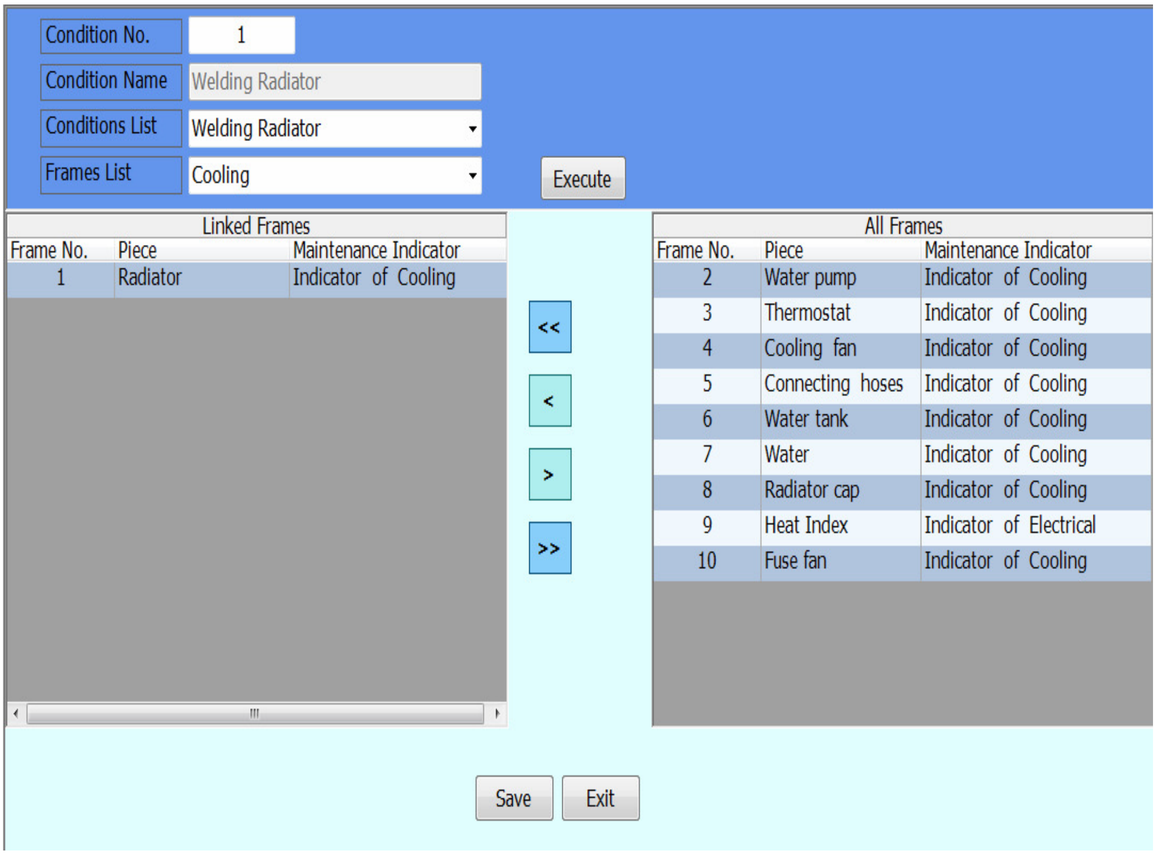


Figure: 5.17: Link Condition and Frame

5.5.3. Show Cases, Conditions and Frames

In figure 5.18 the user can select case name from cases list. A list contains all cases stored before will be displayed. Clicking on Execute button will display a set of conditions related to the frame.

Condition Name	Priority	Piece	Maintenance Indicator	System
Check water temperature sensor	Priority1	water temperature sensor	Indicator of Electrical	Electric & Ignition
check air filter	Priority2	Air Filter	Indicator of Mechanics	Fuel
Check Gasoline	Priority3	Fuel tank	Indicator of Mechanics	Fuel
Check injectors	Priority4	Injection	Indicator of Electrical	Fuel
Check Key time to run the inje...	Priority5	Key time to run the injectors	Indicator of Electrical	Electric & Ignition
Chick Throttle	Priority6	Throttle	Indicator of Mechanics	Fuel
Chick Throttle sensor	Priority7	Throttle sensor	Indicator of Electrical	Fuel
Check the air passage	Priority8	Air cycle	Indicator of Mechanics	Fuel

Figure 5.18: Show Case, Condition and Frame

5.5.4. Find Cases

The user can select a condition from the condition list. A list contains all condition stored before will be displayed. Clicking on Execute button conditions related to the following cases will be displayed: Exact Case, Match Case, No Match (New Case).

This form helps us to correctly deduct and make the right decision to identify the problem.

5.5.4.1. Exact Case

In figure 5.19: All the conditions inserted are equivalent to all conditions stored in the exact case

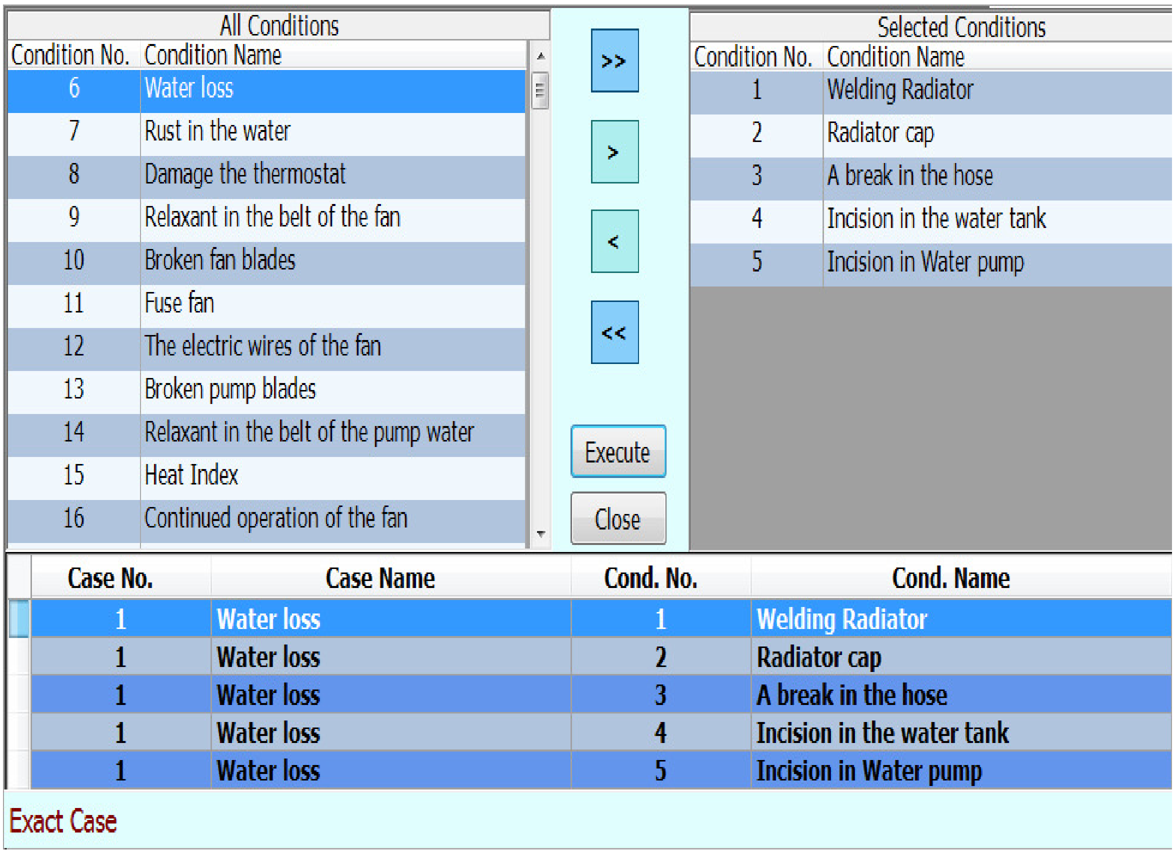


Figure 5.19: Exact Case

5.5.4.2. Match Case

In figure 5.20: All the conditions inserted are available in the same cases

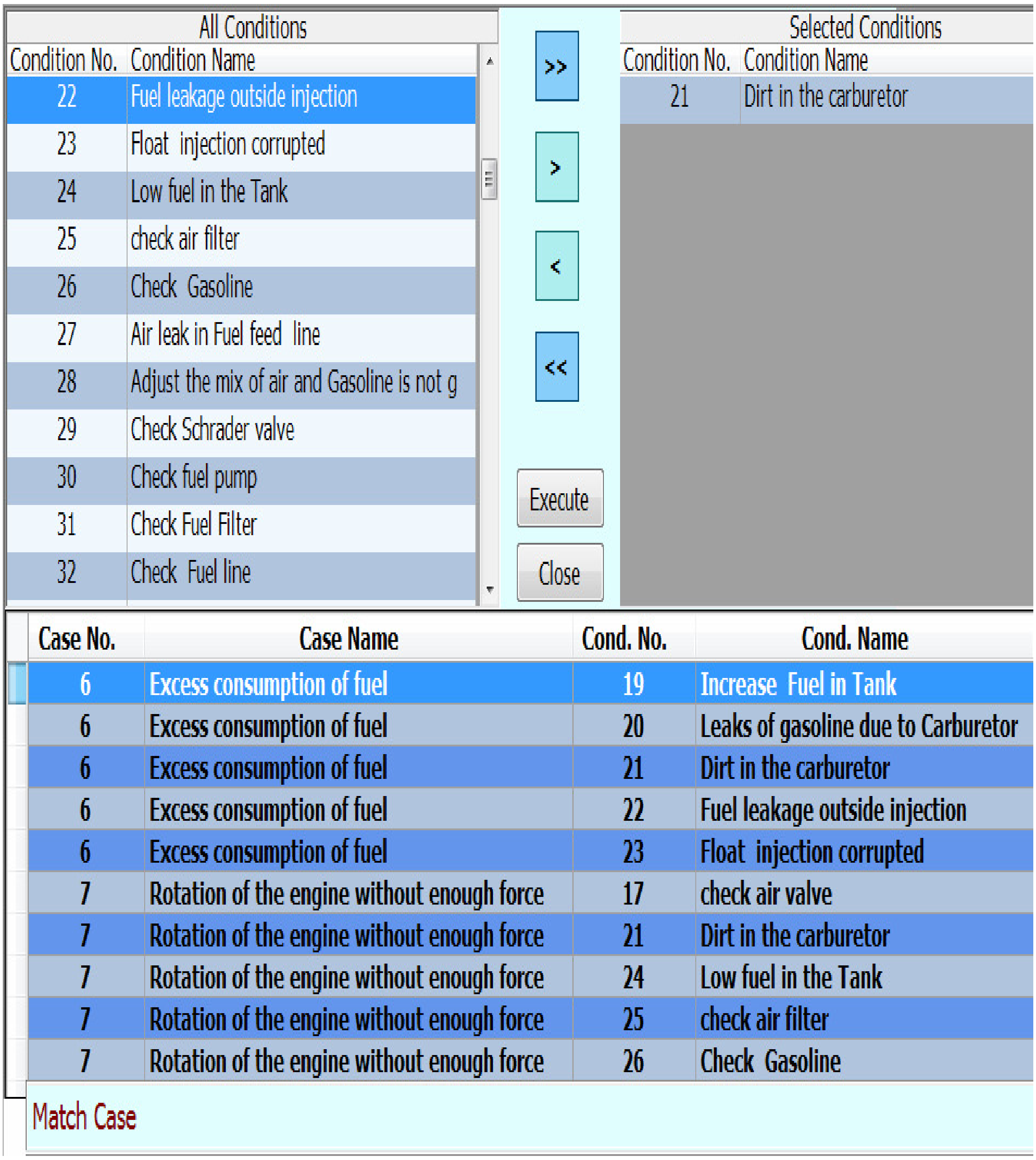


Figure 5.20: Match Cases

5.5.4.3. No Match Case (New Case)

In figure 5.21: All the conditions inserted will be unavailable in the same cases. Conditions can be considered as part of a new case.

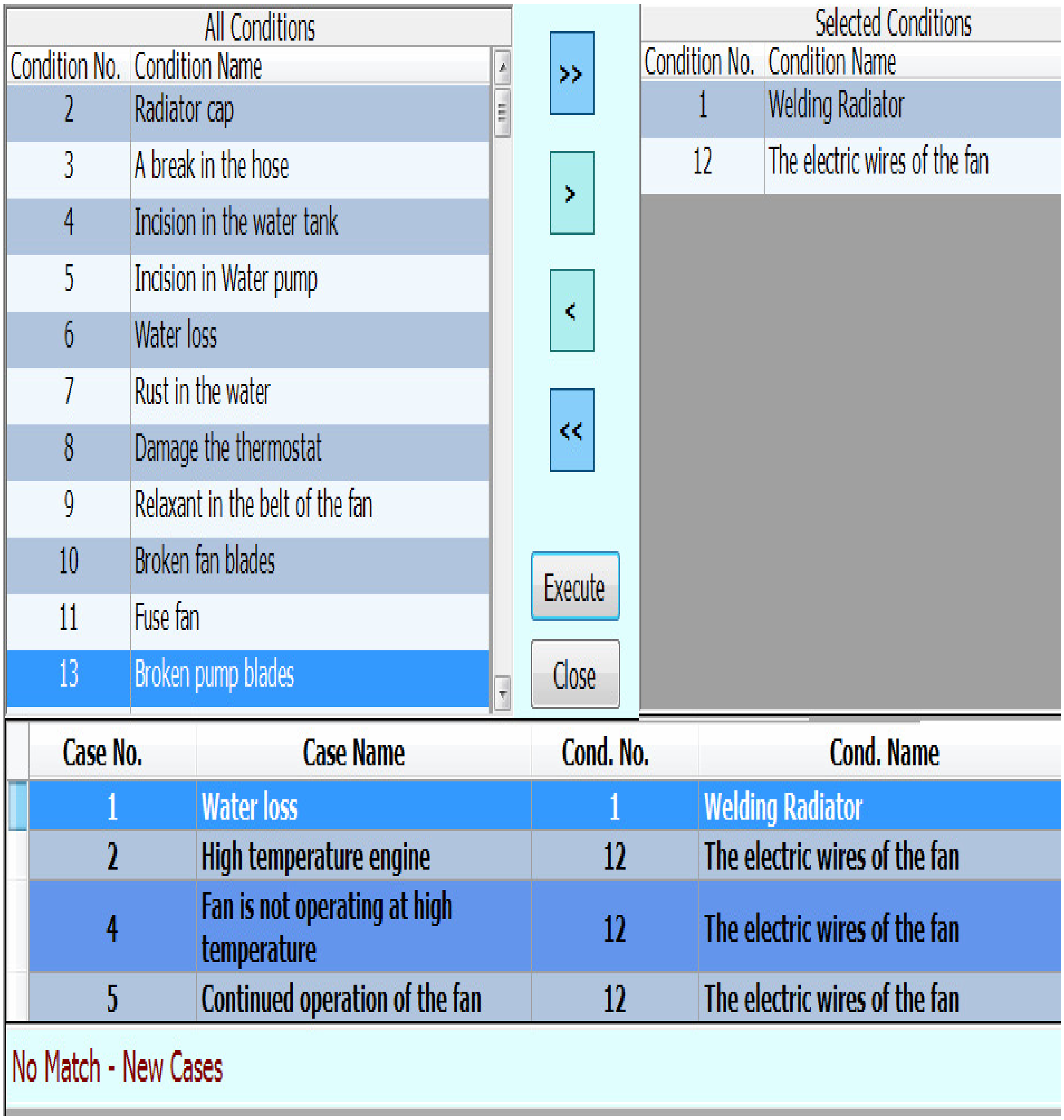


Figure 5.21: New Case

5.6. User Manual

Intelligent Database System is designed to be used by anybody with minimum or lack of programming knowledge.

For downloading and right functioning of The Intelligent Database System for Vehicle Fault Diagnosis the user should have Microsoft Visual Studio 2008 and SQL Server 2005.

Downloading The Intelligent Database System, the main menu will be displayed in order to allow the user to add, edit and update the knowledge.

Main menu which consists of eight forms in two component (Editing Facilities, Explanation facilities) as shown in figure 5.22

Figure 5.22: Main Menu

Editing facilities is used to manage the following: inserting, deleting and updating process for the knowledge base. Editing Facilities consists of four forms which are: edit cases, edit conditions, edit frames, edit rules table as shown in figure 5.23.

Figure 5.23: Editing Facilities

Edit cases form is used to work on cases, which means the user can insert a case to be added to the case base knowledge, delete an existing one, or update existing case as shown in figure 5.24.

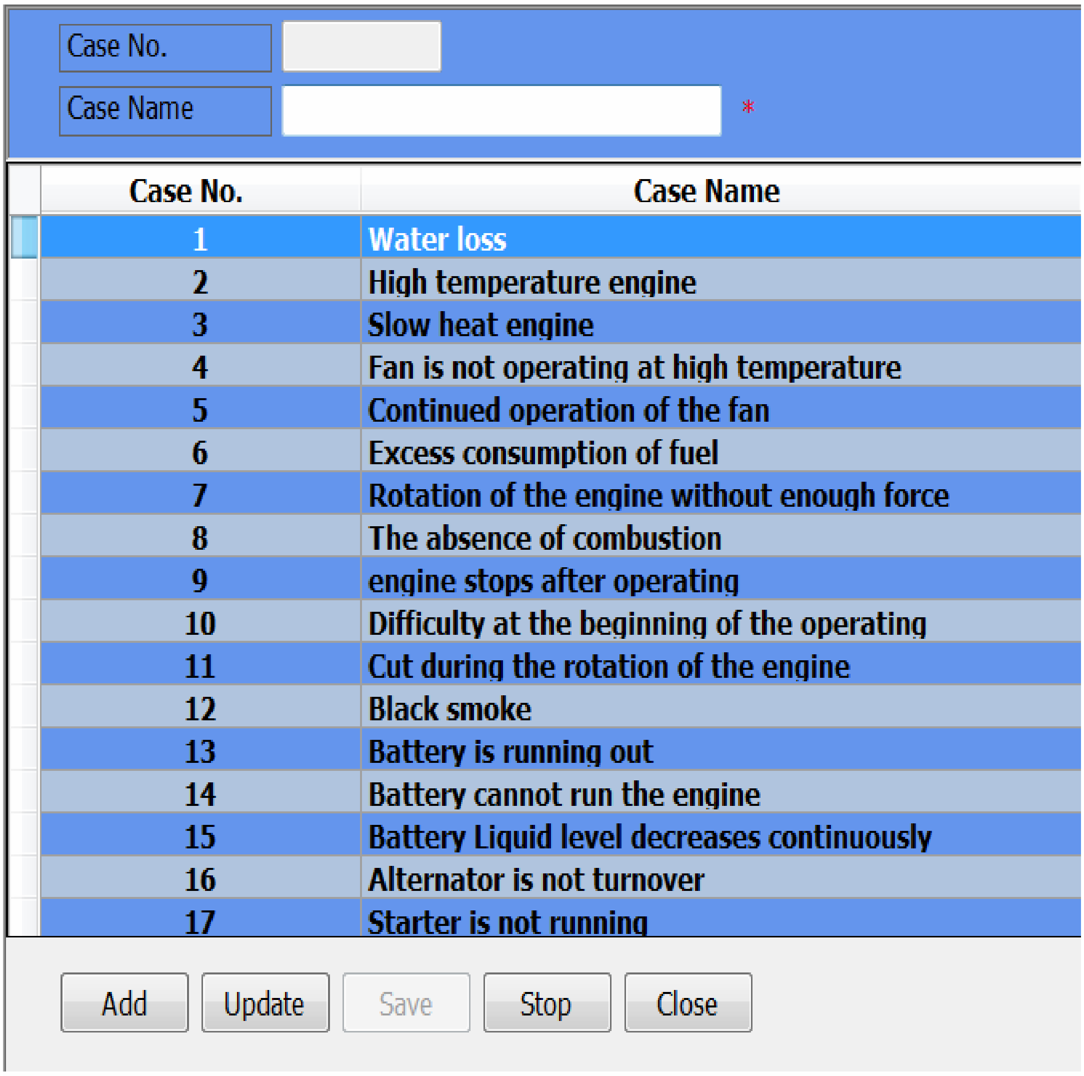


Figure 5.24: Editing Cases

Edit conditions form is used to edit the condition, which means the user can insert a condition to be added to the rule base knowledge, delete one, or update existing condition as shown in figure 5.25

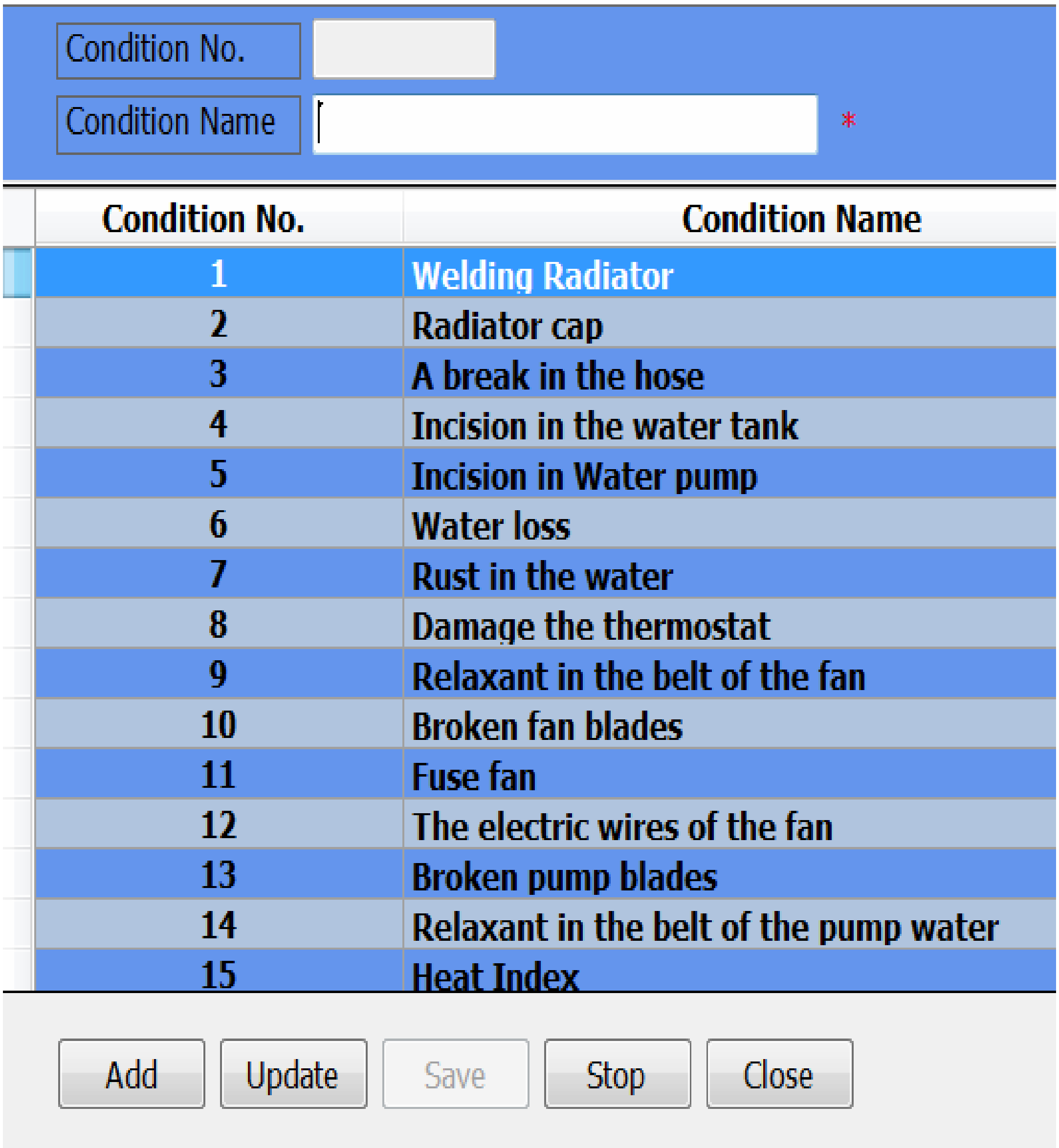


Figure 5.25: Editing Conditions

Edit Frames form is used to edit the frame, which means the user can insert a new part (piece) to be added to the frame base knowledge, delete an existing part, or update an existing one as shown in figure 5.26.

Figure 5.26: Editing Frames

Edit Rules Table form, the user can modify a rule by double clicking on one of the rows. as shown in figure 5.27

Case Name	Condition 1	Condition 2	Condition 3	Case Name
Water loss	Welding Radiator	Radiator cap	A break in the hose	Water loss
Water loss	Incision in the water tank	Incision in Water pump		
High temperature engine	Water loss	Rust in the water	Damage the thermostat	High temperature engine
High temperature engine	Relaxant in the belt of the fan	Broken fan blades	Fuse fan	High temperature engine
High temperature engine	The electric wires of the fan	Broken pump blades	Relaxant in the belt of the pump water	High temperature engine
High temperature engine	Heat Index			
Slow heat engine	Damage the thermostat	Heat Index	Continued operation of the fan	Slow heat engine
Slow heat engine	check air valve	Check water temperature sensor		
Fan is not operating at high temperature	Relaxant in the belt of the fan	Fuse fan	The electric wires of the fan	
Continued operation of the fan	Fuse fan	The electric wires of the fan		
Excess consumption of fuel	Increase Fuel in Tank	Leaks of gasoline due to Carburetor	Dirt in the carburetor	Excess consumption of fuel
Excess consumption of fuel	Fuel leakage outside injection	Float injection corrupted		
Rotation of the engine without enough force	check air valve	Dirt in the carburetor	Low fuel in the Tank	Rotation of the engine without enough force
Rotation of the engine without enough force	check air filter	Check Gasoline	Air leak in Fuel feed line	Rotation of the engine without enough force
Rotation of the engine without enough force	Adjust the mix of air and Gasoline is not good	Check Schrader valve		
The absence of combustion	Check fuel pump	Check Fuel Filter	Check Fuel line	The absence of combustion
The absence of combustion	Check injectors			
engine stops after operating	Check Gasoline	Check coil		
Difficulty at the beginning of the operating	check air valve	Check water temperature sensor	Check Fuel Filter	Difficulty at the beginning of the operating
Difficulty at the beginning of the operating	Check Fuel line	Check injectors	Check Key time to run the injectors	
Cut during the rotation of the	Check water temperature	[illegible]	[illegible]	Cut during the rotation of the

Refresh Close

Figure 5.27: Editing Rules Table

Explanation facilities: Through conditions, cases and frames stored the user can discover new facts and rules that make him able to get and take correct deduction. Editing Facilities consists of four forms which are Link Cases and Conditions, Link Conditions and Frames, Show Cases - Conditions and Frames, and Find Cases. as shown in figure 5.28

Figure 5.28: Explanation facilities

Link Cases and Conditions form: the user can create a link between a case and conditions by entering the case's name in the field cases list. A list will be displayed it contains all cases that were stored before. After selecting the case, a set of conditions stored before will be displayed, by selecting and moving the conditions desired, the link process will be determined. Through the use of arrows priority conditions can be arranged. as shown in figure 5.29

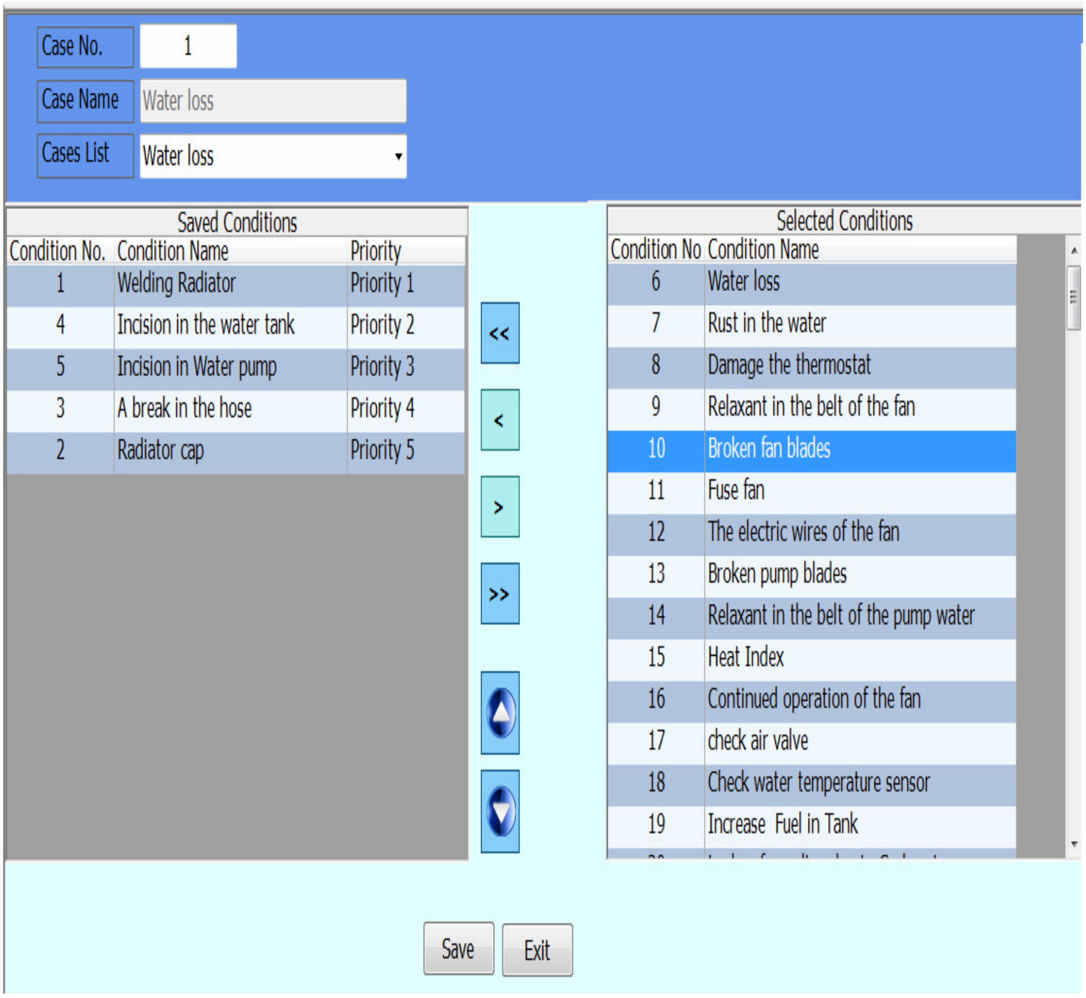


Figure: 5.29: Link Case and Condition

Link Conditions and Frames form: The user can create a link between the conditions and frame by inserting the condition name in the field conditions list. A list will be displayed. It contains all conditions that were stored before. By choosing the frame's name off the field frame list, a list contains all frames stored before will be displayed. A list of parts (pieces) will be displayed when the user clicks on Execute button. The next step is selecting the part and clicking the save button the link process will be determined.

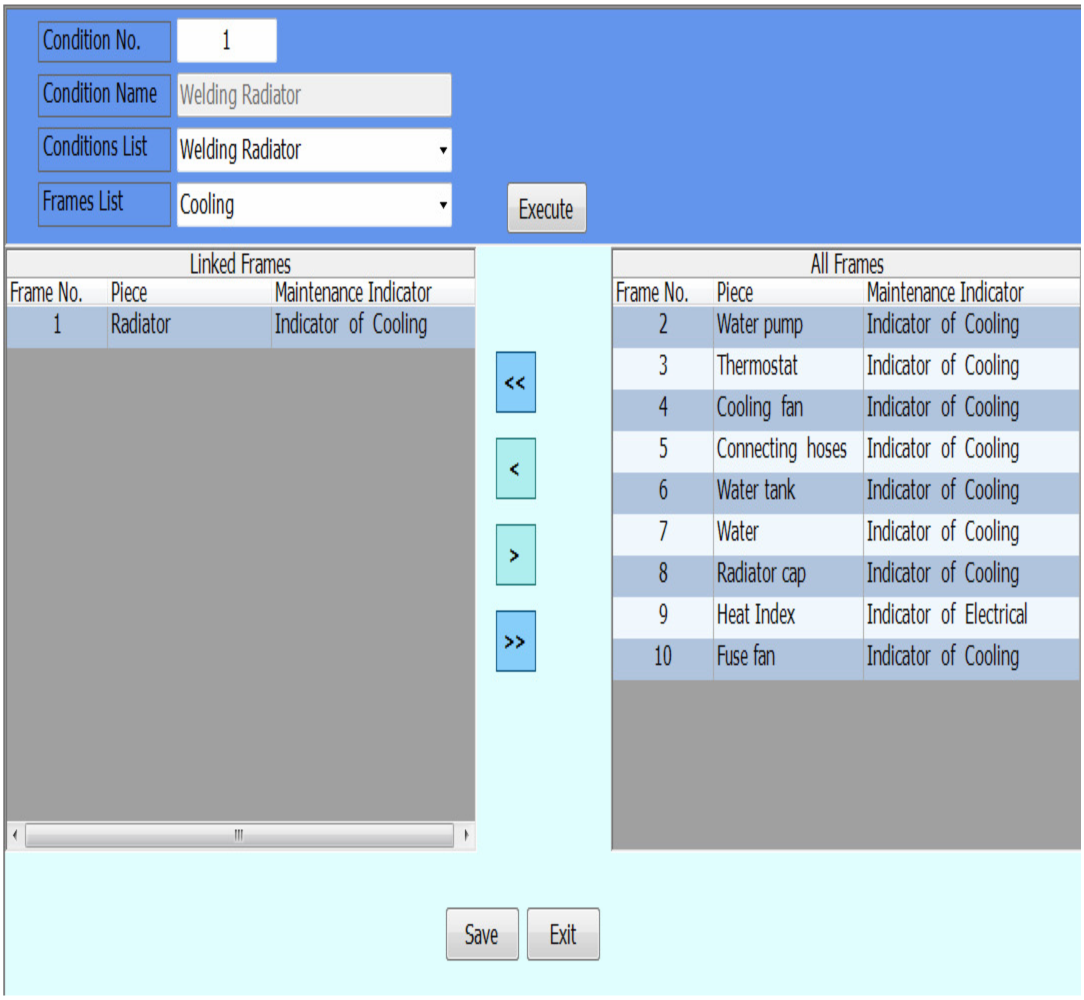


Figure: 5.30: Link Condition and Frame

Show Cases - Conditions and Frames Form: The user can select a case name from cases list. A list contains all cases stored before will be displayed. Clicking on Execute button will display a set of conditions related to the frame.

Condition Name	Priority	Piece	Maintenance Indicator	System
Check water temperature sensor	Priority1	water temperature sensor	Indicator of Electrical	Electric & Ignition
check air filter	Priority2	Air Filter	Indicator of Mechanics	Fuel
Check Gasoline	Priority3	Fuel tank	Indicator of Mechanics	Fuel
Check injectors	Priority4	Injection	Indicator of Electrical	Fuel
Check Key time to run the inje...	Priority5	Key time to run the injectors	Indicator of Electrical	Electric & Ignition
Chick Throttle	Priority6	Throttle	Indicator of Mechanics	Fuel
Chick Throttle sensor	Priority7	Throttle sensor	Indicator of Electrical	Fuel
Check the air passage	Priority8	Air cycle	Indicator of Mechanics	Fuel

Figure 5.31: Show Case, Condition and Frame

Find Cases: The user can select a condition from the conditions list. A list contains all conditions stored before will be displayed. Clicking on Execute button, conditions related to the following cases will be displayed: Exact Case, Match Case, No Match (New Case).

Figure 5.32: Find Case

5.7. Advantages of Intelligent Database System

a) The system will perform better with each new case.

b) The system is accurate in its answers.

c) Ease of modification to the knowledge base.

d) Providing advice with no requirement for a human expert.

e) Ease of use by the user.

Chapter Six

Conclusion and Future Work

Chapter Six

Conclusion and Future Work

6.1. Conclusion

In this thesis, Design and Implementation of Intelligent Database System for Vehicle Fault Diagnosis, the following points can be concluded:

1) The implementation of knowledge base depends on knowledge representation forms of the, usually, represented knowledge in many different forms. In this thesis knowledge base has been represented in three forms which are rule base, case base and frame base.

2) Intelligent database system was designed for the normal user, who doesn't know programming, but can only add knowledge.

3) The end user can use all editing facilities like inserting, deleting and updating of knowledge base.

4) The system implementation in the diagnosis of fault vehicle and the results of this system were matched with the decisions taken by the vehicle mechanics.

6.2. Future Work

There are many of the future works that can be made depending on the system, which can be summarized as follows:

1) The system can be used as an application software for any similar application .

2) The system will be able to provide a rich interactive forum to enable the users to ask questions and receive answers.

References

Abbass, H.A. , Sarker, R.A. & Newton C.S. (2002). *Data Mining: A Heuristic Approach*, Idea Group Publishing, London - United Kingdom.

Alahmar, M.A. (2010). Rule based expert system for selecting software development methodology, ***Journal of Theoretical and Applied Information Technology***, 2005 - 2010 JATIT & LLS. All rights reserved, vol. 19, no. 2, pp. 143-148.

Ana, M. & Jose, A. (2007). A General ontology for intelligent database, ***International Journal of Computers, ,*** vol. 1, no 3, pp.102-108**.**

Antonsen, K.M. & Viazzi, S. (2006). IT 3706 Knowledge Representation Lecture 4: Rule-Based Systems, Norwegian University of Science and Technology, September 20, 2006.

Arman, N. (2006). Parallel algorithms for the generalized same generation query in deductive databases, ***Journal of Digital Information*** **Management**, Vol. 4, No. 3, , pp. 192- 196, ISSN 0972-7272.

Chen, T. , Wu, J.K. & Takagi, M. (1991). Frame representation of ecological models in forestry planning, pp. 816-820, University of Tokyo, Japan.

Coenen, F. (1998). Verification and validation issues in expert and database systems: the expert systems perspective, ***Database and Expert Systems Applications***, Liverpool, England.

El-Helw, A.F. & Aly, H.H (2004).An Intelligent database application for the semantic web, *CSITeA-04*, Cairo - Egypt, 2004.

Girratano, R. (1998). *Expert systems, principles and programming*. Boston: PWS Publishing,.

Gonzalez A.J. and Dankel D.D. (1993). *The Engineering of knowledge-based systems: theory and practice.* Englewood Cliffs, NJ: Prentice-Hall.

Jafari, M. , Akhavan, P. & Akhtari, M. (2011). Exploration of knowledge acquisition techniques in tunnel industry: the case study of Iran tunnel association, ***International Journal of Business and Management ,*** vol. 6, no 8, pp.245-254, August 2011.

Jain, B.M. , Jain, A. & Srinivas, M.B (2008). A Web based expert system shell for fault diagnosis and control of power system equipment, ***International Conference on Condition Monitoring and Diagnosis***, pp.1-5, April 21-24, 2008, Hyderabad, India.

Joyner, D. , Nguyen, M.V. & Cohen, N. (2011). *Algorithmic graph theory*, (6th ed.)

Luger, G.F. (2005). *Artificial intelligence: structures and strategies for complex problem solving methodologies ,* (5th ed.) Addison-Wesley.

Nihalani, N. , Silakari, S. & Motwani, M. (2009). Design of intelligent layer for flexible querying in databases, ***International Journal on Computer Science and Engineering***, vol. 1, no 2, pp.30-39.

Nilsson, N.J. (1971). *Problem solving methods in artificial intelligence*, New York: McGraw-Hill.

Owaied, H.H. & Abu-Arr'a , M.M. (2007). Functional model of human System as knowledge Base System, ***The 2007 International Conference on Information & Knowledge Engineering***, pp.158-161, June 25-28,2007.

Owaied, H.H. , Abu-Arr'a, M.M. & Farhan, H.A. (2010). An Application of knowledge based system, ***IJCSNS International Journal of Computer Science and Network Security,*** vol. 10, no 3, pp.208-213.

Parsaye, K. , Chignell, M. , Khoshafian, S. & Wong, H., (1989). *Intelligent databases: object-oriented, deductive hypermedia technologies*, New York, John Wiley & Sons, 1989.

Pezzulo, G. (2011). Grounding procedural and declarative knowledge in sensorimotor anticipation, ***Blackwell Publishing Ltd***, vol. 26, no 1, pp.78-114.

Potter, C. & Negnevitsky, M. (2003). An Expert system application for hydro electric generator scheduling in Tasmania. ***Journal of Electrical & Electronics Engineering***, vol. 22, no 3, pp.167-171, Australia

Ralston, A. & Reilly, E. (1993). *Encyclopedia of computer science and engineering*, Thomson Learning, 1993

Rattanaprateep, C. & Chittayasothron, S. (2006). A Frame based object-relational database expert system architecture and implementation , ***Proceedings of the 5th WSEAS Int. Conf. on Artificial Intelligence- Knowledge Engineering and Data Bases,*** pp.327-332, February 15-17, 2006.

Reisbeck, C.K. , & Schank, R.C. (1989). Inside Case-Based Reasoning. Lawrence Erlbaum Associates, Hillsdale, NJ, US.pp.423.

Russel, S. & Norvig, B. (2003). *Artificial intelligence a modern approach*, (2nd edn.) Pearson Education.

Salem, A.M. , Roushdy, M. & Hodhod, R.A. (2005). A Case based expert system for supporting diagnosis of heart diseases, ***AIML Journal*** , vol. 5, no 1, pp. 33- 39.

Saritas, I. , Allahverdi, N. & Sert, I.U. (2003). A Fuzzy expert system design for diagnosis of prostate cancer, ***International Conference on Computer Systems and Technologies – CompSysTech.***

Sarma, S.K. , Singh, K.R. & Singh, A. (2010). An Expert system for diagnosis of diseases in rice plant, ***International Journal of Artificial Intelligence ,*** vol. 1, no 1, pp. 26-31, India.

Sharma, A.K. , Kumar, C. , Mustafa, K. & Kumar, A. (2003). A Fuzzy frame based expert shell, ***Proceedings: National Workshop on IT Services and Applications (WITSA).***

Sirikumara, P. (2006). *Test result archiving and diagnosis systems*, (Master's Thesis in Computing Science), UMEA University , Umea, Sweden.

Stockburger, D.W. (2007). *Learning focused,* (3rd edn.) Atomic Dog.

Thuraisingham, M.B. (2010). Secure query processing in intelligent database management system, The MITRE Corporation, Burlington Road, Bedford, MA.

Tomic´, B. , Jovanovic´, J. & Devedzˇic, V. (2006). JavaDON: an open-source expert system shell ,***Expert Systems with Applications* ,** pp.595–606.

Touch, J.D. (1994). Telecommunications issues of intelligent database management for ground processing systems in the EOS Era, ***Reprinted from Telematics and Informatics***, vol. 11, no. 4, pp. 319-332, 1994

Villa, F. , Ceroni, M. & Krivov, S. (2006). Intelligent databases assist transparent and sound economic valuation of ecosystem services, ***Environ Manage***, pp.887-899 ,Springer Science+ Business Media, 29 October 2006

Zadeh, L.A. (1965). Fuzzy Sets, ***Information and Control***, vol. 8, no 3, pp.338-353, June 1965.