iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** VI **Month of publication:** June 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

An Efficient Use of Intellisense through ASP.NET in SQL Reporting On Webpage

Ankur Sharma[1], Rachna Dahiya[2]
[1] M.tech Scholar, [1]Department of Information Technology
[2] Assistant professor, [2]Department of Computer Science Engineering
GIET Sonepat, Haryana, India

Abstract—Database systems are designed to manage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results. Because information is so important in most organizations, computer scientists have developed a large body of concepts and techniques for managing data. In database we can write queries to extract data. But in our research we used stored procedures for better efficiency. The efficiency increases when we use stored procedure as they are stored in executed form and we run the report they give output but in case of query they executed everytime we run the report. The main purpose of this research is to help in report generation when we fetch data from database to view in the form of report. In the normal scenario when we want to generate report then there may be some parameters on the basis of which we fetch data. Without parameter the report is generated for the whole data at a time. Generally if we want to select data in parameter and the list of data is very large may be in thousands then at that time we have to first search the whole list to find that data. It will become very difficult and time consuming task for us to find the required data. By using this tool and technique which we have done in research we provide intellisense in searching which makes it easier to search. We just need to type alphabet of the required parameter and it will give you all the results related to that alphabet. Normally we can generate reports in SQL server data tool but by using Asp.net we can see or generate report on the webpage. This will help to display the report publically on the webpage.
Keywords—Database system, manipulation, unauthorized, queries, fetch, efficiency, intellisense .

I. INTRODUCTION

A database is any collection of related data. A database is a persistent, logically coherent collection of inherently meaningful data, relevant to some aspects of the real world. Database Management Systems (DBMSs) are complex, mission-critical software systems. Today's DBMSs embody decades of academic and industrial research and intense corporate software development. Database systems were among the earliest widely deployed online server systems and, as such, have pioneered design solutions spanning not only data management, but also applications, operating systems, and networked services. The early DBMSs are among the most influential software systems in computer science, and the ideas and implementation issues pioneered for DBMSs are widely copied and reinvented. For a number of reasons, the lessons of database systems architecture are not as broadly known as they should be. First, the applied database systems community is fairly small. Since market forces only support a few competitors at the high end, only a handful of successful DBMS implementations exist. The community of people involved in designing and implementing database systems is tight: many attended the same schools, worked on the same influential research projects, and collaborated on the same commercial products. Second, academic treatment of database systems often ignores architectural issues. Textbook presentations of database systems traditionally focus on algorithmic and theoretical issues — which are natural to teach, study, and test — without a holistic discussion of system architecture in full implementations. In sum, much conventional wisdom about how to build database systems is available, but little of it has been written down or communicated broadly.

A. Objective Of The Research

The objective of this research is to generate the report on the webpage and reduce the time of user in selecting the parameter using intellisense. The research tries to solve the following problem:How much importance needs to be given in showing the report on the webpage and how we can easily search the parameter value from the large amount of data?

II. RESEARCH METHODOLOGY

In order to achieve the research objective and try to answer question the following steps has been followed:

©IJRASET 2015: All Rights are Reserved

Study the literature related to database processes.

Study the literature related to Queries for database and study about the different kinds of stored procedures that can be made in database written in SQL

Develop a report in SQL server data tool by extracting data from the database using the stored procedures.

Develop a platform in Visual Studio using Ajax control toolkit in Asp.net for showing the implementation of the intellisense used for parameters in reports.

Implementation of the model along with the test data to show what can be leveraged out of this model.

III. INTRODUCTION TO DATABASE, STORED PROCEDURES AND REPORTS

A. Introduction

Databases and database technology are having a major impact on the growing use of computers. It is fair to say that databases play a critical role in almost all areas where computers are used, including business, engineering, medicine, law, education, and library science, to name a few. The word database is in such common use that we must begin by defining a database. Our initial definition is quite general. A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book, or you may have stored it on a diskette, using a personal computer and software such as DBASE IV or V, Microsoft ACCESS, or EXCEL. This is a collection of related data with an implicit meaning and hence is a database.The preceding definition of database is quite general; for example, we may consider the collection of words that make up this page of text to be related data and hence to constitute a database. However, the common use of the term database is usually more restricted. In other words, a database has some source from which data are derived, some degree of interaction with events in the real world, and an audience that is actively interested in the contents of the database. A database may be generated and maintained manually or it may be computerized. The library card catalog is an example of a database that may be created and maintained manually. A computerized database may be created and maintained either by a group of application programs written specifically for that task or by a database management system. A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is hence a general-purpose software system that facilitates the processes of defining, constructing, and manipulating databases for various applications. Defining a database involves specifying the data types, structures, and constraints for the data to be stored in the database. Constructing the database is the process of storing the data itself on some storage medium that is controlled by the DBMS. Manipulating a database includes such functions as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld, and generating reports from the data. It is not necessary to use general-purpose DBMS software to implement a computerized database. We could write our own set of programs to create and maintain the database, in effect creating our own special-purpose DBMS software. In either case—whether we use a general-purpose DBMS or not—we usually have to employ a considerable amount of software to manipulate the database. We will call the database and DBMS software together a database system.

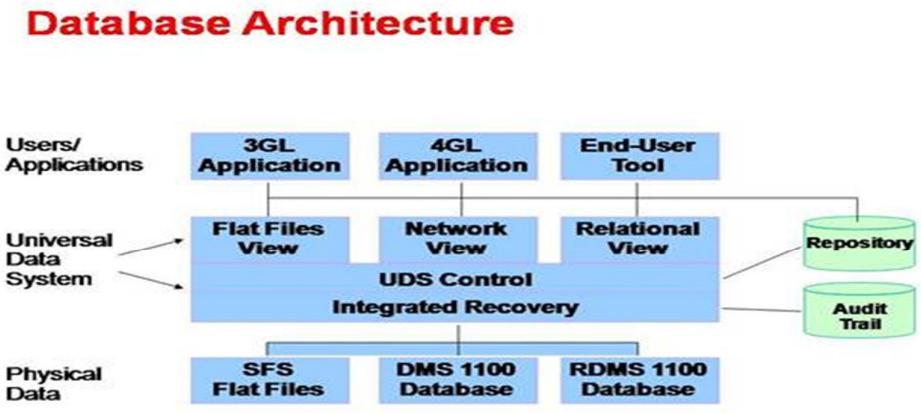


Fig. 1 Database Architecture

1) Characteristics Of The Database Approach: A number of characteristics distinguish the database approach from the traditional approach of programming with files. In traditional file processing, each user defines and implements the files needed for a specific application as part of programming the application. In the database approach, a single repository of data is maintained that is defined once and then is accessed by various users. The main characteristics of the database approach versus the file-processing approach are the following.

a) Self-Describing Nature Of A Database System : A fundamental characteristic of the database approach is that the database system contains not only the database itself but also a complete definition or description of the database structure and constraints. This definition is stored in the system catalog, which contains information such as the structure of each file, the type and storage format of each data item, and various constraints on the data. The information stored in the catalog is called meta-data, and it describes the structure of the primary database.

b) Insulation Between Programs And Data, And Data Abstraction: In traditional file processing, the structure of data files is embedded in the access programs, so any changes to the structure of a file may require changing all programs that access this file. By contrast, DBMS access programs do not require such changes in most cases. The structure of data files is stored in the DBMS catalog separately from the access programs. We call this property program-data independence. In object-oriented and object-relational databases, users can define operations on data as part of the database definitions. An operation (also called a function) is specified in two parts. The interface (or signature) of an operation includes the operation name and the data types of its arguments (or parameters). The implementation (or method) of the operation is specified separately and can be changed without affecting the interface. User application programs can operate on the data by invoking these operations through their names and arguments, regardless of how the operations are implemented. This may be termed **program-operation** independence. The characteristic that allows program-data independence and program-operation independence is called data abstraction. A DBMS provides users with a conceptual representation of data that does not include many of the details of how the data is stored or how the operations are implemented. Informally, a data model is a type of data abstraction that is used to provide this conceptual representation. The data model uses logical concepts, such as objects, their properties, and their interrelationships, that may be easier for most users to understand than computer storage concepts.

c) *Support Of Multiple Views Of The Data:* A database typically has many users, each of whom may require a different perspective or view of the database. A view may be a subset of the database or it may contain virtual data that is derived from the database files but is not explicitly stored. Some users may not need to be aware of whether the data they refer to is stored or derived. A multiuser DBMS whose users have a variety of applications must provide facilities for defining multiple views. A second user, who is interested only in checking that students have taken all the prerequisites of each course they register for.

*d) Sharing Of Data And Multiuser Transaction Processing***:** A multiuser DBMS, as its name implies, must allow multiple users to access the database at the same time. This is essential if data for multiple applications is to be integrated and maintained in a single database. The DBMS must include concurrency control software to ensure that several users trying to update the same data do so in a controlled manner so that the result of the updates is correct. For example, when several reservation clerks try to assign a seat on an airline flight, the DBMS should ensure that each seat can be accessed by only one clerk at a time for assignment to a passenger. These types of applications are generally called on-line transaction processing (OLTP) applications. A fundamental role of multiuser DBMS software is to ensure that concurrent transactions operate correctly.

B. Processing Stored Procedures

When orking with SQL Server stored procedures, you need to be aware of several aspects of how stored procedures work. These aspects include how stored procedures are created, how they are stored, and how they are processed. In this, you will learn about the following:

How SQL Server creates stored procedures
How SQL Server processes stored procedures
SQL Server execution plans

*1) How SQL Server Creates Stored Procedures***:** Unfortunately, despite the title of this section, SQL Server doesn't create stored procedures for you. You have to plan out how the stored procedure will work, write the code, and then debug the code. Nevertheless, when you finally have your code written and execute the CREATE PROCEDURE statement, SQL Server has to go through a couple of steps before the code is actually created as a stored procedure. When CREATE PROCEDURE is

©IJRASET 2015: All Rights are Reserved

executed, SQL Server first parses the statement to ensure that all statements are syntactically correct. If the server encounters any sort of syntax error, the creation of the procedure is cancelled and an error is returned to the user. If there are no syntax errors, SQL Server stores the text of the stored procedure in the syscomments table in the database that the user who created the stored procedure was using at the time CREATE PROCEDURE was executed. Nothing else happens until the stored procedure is actually executed. After the stored procedure is run for the first time, SQL Server prepares the code contained in the procedure to be executed.

2) *How SQL Server Processes Stored Procedures***:** When you first execute a stored procedure, the SQL Server query processor reads the code of the stored procedure from the syscomments table and begins the deferred name resolution process.

3) *Programming Stored Procedure Rules:* When you are developing an application that relies on SQL Server stored procedures, you must take into account a number of rules.
Several rules must be dealt with when creating stored procedures. These rules define the way in which the stored procedure itself is created and how the stored procedure can interact with other objects in the database. The following is a list of the major things you must keep in mind when creating stored procedures:
The CREATE PROCEDURE statement must be the first statement in a SQL Server batch. If the procedure is not the first statement, SQL Server returns an error to the user and fails to create the procedure.
Stored procedures can execute any SQL statement available to the creator, except for the following statements: CREATE DEFAULT, CREATE TRIGGER, CREATE PROCEDURE, CREATE VIEW, and CREATE RULE.
Any database object, except for those SQL statements listed earlier, can be created and used within a stored procedure.
Stored procedures can create and reference temporary tables.
If you create a local, or private, temporary table inside a stored procedure, that table is in the database only for the use of the stored procedure that created it. When the stored procedure has completed its execution, the temporary table is dropped from the database.
If you create a stored procedure that executes a second stored procedure, all objects created in the first stored procedure are available to the second stored procedure. This includes both private and public temporary tables.
If you execute a remote stored procedure that modifies data on a remote server, those changes cannot be rolled back. This is because remote stored procedures do not take place as a part of the local transaction.
SQL Server places no logical limit on the number of internal variables that can be defined in a stored procedure. The amount of available memory determines how many variables can be used.

4) Why Stored Procedures Are Faster: To understand why stored procedures are faster than direct SQL commands, you need to understand how SQL Server processes the commands that it receives. When you submit any command for SQL Server to execute, it goes through the following stages:
The command is parsed for syntax. Any commands that are syntactically incorrect are rejected.
The command is then translated into an internal format known as a sequence tree or query tree.
The command is optimized based on estimated performance costs, and an execution plan is generated from the sequence tree that contains all the necessary steps to check constraints and carry out the command.
The command is executed.
If you send direct SQL statements one at a time from your application, then each statement requires all four stages for each single statement. Contrary to common l. belief, the execution plan is not saved with the stored procedure when it is created. For a stored procedure, stages 1 and 2 happen only once, when you save the stored procedure. Step 3 occurs only if the execution plan is not already in the memory cache. Subsequent executions only need step 4 (execution).

C. Reports In Sql Server Data Tool

In this we explain how to create a SSRS report using a SQL parameterized Stored Procedure object. The main purpose of this article to reduce the extra effort of the SQL developer during various types of report extraction. First we add the report in the project. Then we attach required stored procedure which is used in the report to extract the data and apply the formats i.e matrix or tablix form and some other options which are required. Then we add the required datasets which are used to extract the data in the reports. After that the parameters for the reports are added and then preview it to generate the report.

IV. RESULT AND ANALYSIS

©IJRASET 2015: All Rights are Reserved

A. Implementation Platform

The implementation of the model here includes number of steps like creating stored procedures, execute sql scripts. For all these steps, Microsoft SQL Server 2012 has been used for the implementation. The tool used for implementation is SSMS (like SQL Server Management Studio). SSMS is a great tool to work and is very intuitive and easy of work with. The database server has 2GB ram with 500GB disk space providing a decent machine for this implementation. The database called 'AdventureWorks' is used for the implementation of the model. Then reports are generated in SSDT (SQL Server Data Tool) by using the stored procedures. Then these reports are generated on the webpage including intellisense property by using ASP.Net.

B. Test And Results

This section presents the result screenshots of the tests being performed on the source data and Asp.net implementation.

Source data Implementation by Stored Procedures

Screen Shot 1:

Figure 2 screen shot shows the stored procedure written with one parameter of name in it. The stored procedure named here is sp_ProductIntellisense. This stored procedure extract data from the database AdventureWorks.

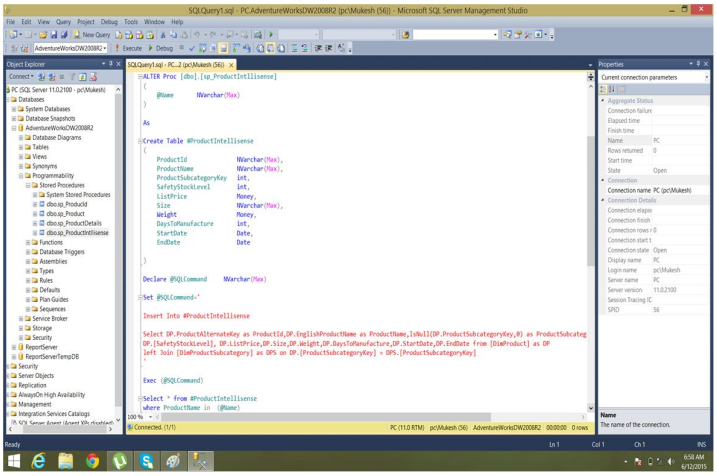


Figure 2. Results #1.

Screen Shot 2:

Figure 3 screen shot shows the stored procedure executed. In the bottom of the screen we can check the result of the stored procedure by selected parameter.

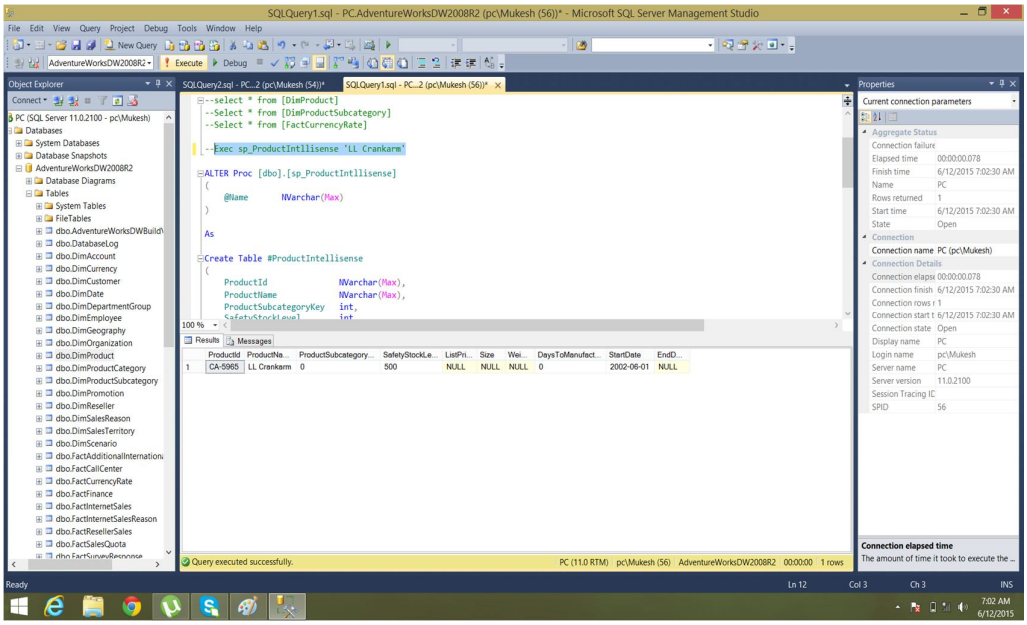


Figure 3. Results #2.

Screen Shot 3 :

Figure 4 screenshot shows the report when it is generated in SQL server data tool

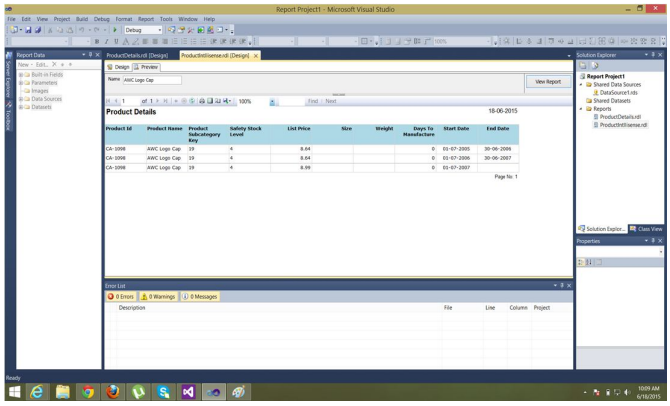


Figure 4. Results #3.

Reports Shown on Webpage with Intellisense

The following screen shots are related to the implementation by which reports are generated on the webpage with intellisense property. The screen shot shows how the coding done in C# to bind the report to the webpage and the use of Ajax Control Toolkit for Intellisense property.

Screen shot 4:

Figure 5 screenshot shows the webpage where report will be generated. In this user has typed one alphabet in the textbox and due to intellisense property all the data related to that alphabet is showing in the list.

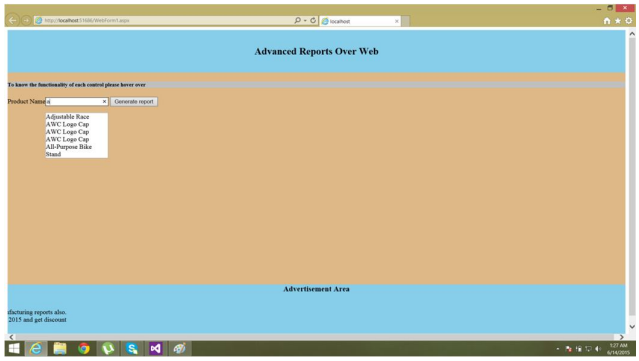


Figure 5. Results #4.

Screen Shot 5:

Figure 6 screenshot shows the report generated on the webpage by selecting any parameter. The required parameter which is selected has the following data which is visible on the report.

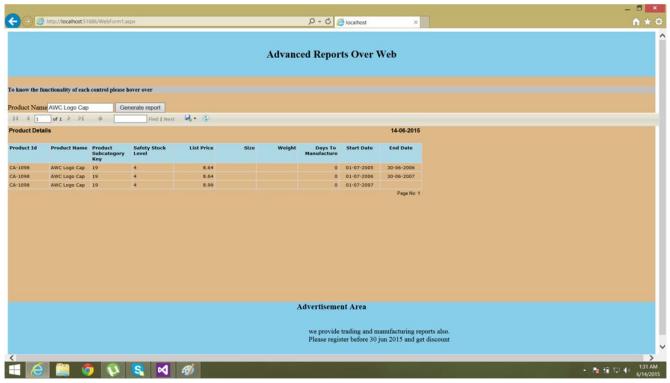


Figure 6. Results #5

V. RESEARCH CONTRIBUTIONS AND POSSIBLE FUTURE WORK

This chapter summarizes the research contributions of this thesis and identifies the possible future scope.

A. Research Contributions

The research contributions of this thesis are given below:

The literature about the database concepts is reviewed. An in depth analysis of topics such as Stored Procedures and identifying the myths about database is discussed.

Then, the architecture of a typical database is explained and also presented the flow of processes of every phase in database.

The implementation is done in SQL Server 2012 server and also presented the screen shots with the test data. The implementation of this model presented how the data will look like and what feature it can provide.

The reports are generated in SQL Server data tool 2012 with parameters. These reports are used to see the data which is extracted by stored procedure from database.

The reports are then bind with the webpage with the help of Asp.Net and made publically available on web.

Ajax Control Toolkit is used to provide property of intellisense which is used in searching when the data is very large.

The test results and the implementation of the research model shows the tip of the iceberg of what we can be done with this approach. Using this model, one can add many more functionalities to the reports for some complex data with multiple parameters.

B. Future Work

The research contribution can be enhanced by making reports with more than one parameter. Some extra work can also be done inorder to select multiple data in the parameter by using intellisense. Dashboards can also be formed by reports in which data is shown in visualized forms in the form and graphs. As already been discussed, this model is just the tip of the iceberg and there is so many ways the research can be extended. It can be extended to implement reports with multiple parameters with multiple data and also making dashboards from the reports.

REFERENCES

[1] Chapple, Mike. SQL Fundamentals. Databases. About.com. Retrieved 2009-01-28.

[2] Structured Query Language (SQL). International Business Machines. October 27, 2006. Retrieved 2007-06-10.

[3] Raul F. Chong, Michael Dang, Dwaine R. Snow, Xiaomei Wang (3 July 2008)."Introduction to DB2". Retrieved 17 March 2013.. This article quotes a development time of 5 years involving 750 people for DB2 release 9 alone

[4] Codd, E.F. (1970). A Relational Model of Data for Large Shared Data Banks. In:Communications of the ACM 13 (6): 377–387.

[5] Beynon-Davies P. (2004). Database Systems 3rd Edition. Palgrave, Basingstoke, UK.ISBN 1-4039-1601-2

[6] R. KIMBALL, A dimensional modeling manifesto, J. Data Base Mgmt. Syst., 10 (1997), pp. 58-70.

[7] R.Halder and A.Cortesi, Abstract Interpretation of Database Query Languages. COMPUTER LANGUAGES, SYSTEMS & STRUCTURES, vol. 38(2), pp. 123--157, Elsevier Ed. (ISSN 1477-8424)

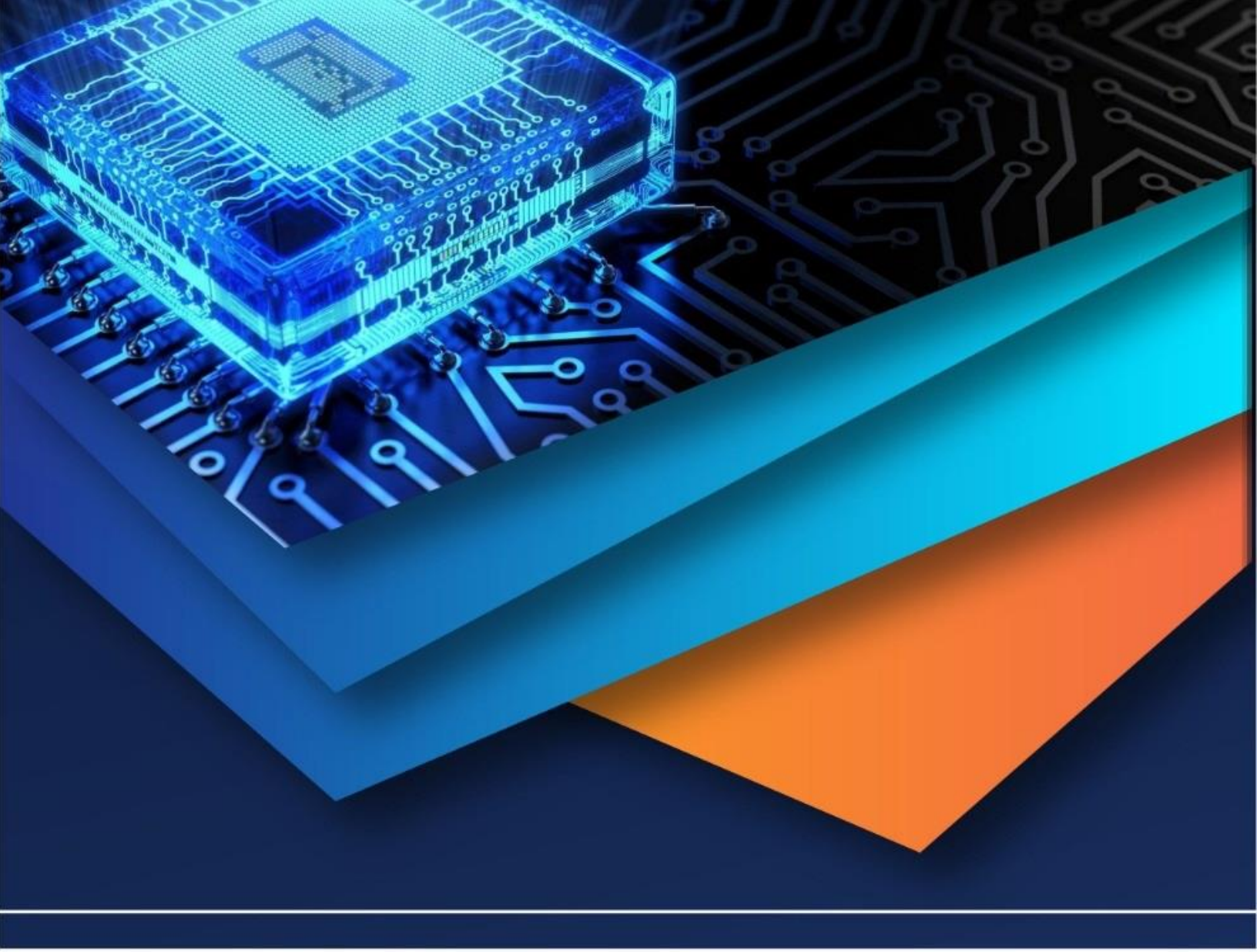
[8] Development of an object-oriented DBMS; Portland, Oregon, United States; Pages: 472 – 482; 1986; ISBN 0-89791-204-7

[9] Lightstone, S.; Teorey, T.; Nadeau, T. (2007). Physical Database Design: the database professional's guide to exploiting indexes, views, storage, and more. Morgan Kaufmann Press.ISBN 0-12-369389-6.

[10] Teorey, T.; Lightstone, S. and Nadeau, T. Database Modeling & Design: Logical Design, 4th edition, Morgan Kaufmann Press, 2005. ISBN 0-12-685352-5

[11] "ASP.NET Web Page Syntax Overview". Microsoft .NET Framework. Microsoft. Retrieved 2010-11-22.[12] Stephen Walther: ASP.NET 3.5 Unleashed, December 28, 2007, Sams Publishing, ISBN 0-672-33011-3 ISBN 0-672-33011-3

©IJRASET 2015: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)