**ACKNOWLEDGEMENT**

We express our deep sense of gratitude to

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY,** Bangalore

which has provided us an opportunity to fulfill our desires in reaching our goal.

* We thank our principal **Dr. H.C Nagaraj** for providing the lab facilitieswe needed for the completion of our project.
* We are grateful to our H.O.D, **Dr. Sanjay H.A** who extended his support towards our project.
* We would like to thank our assistant & our lecturer

**Mr. Adithya Shastry sir** for helping us in betterment of our project.

* Finally we would like to thank all our friends who have supported me us in this regard.

**INTRODUCTION:**

**DATABASE MANAGEMENT SYSTEM**

**Database:** A database is a collection of related data. Defining a database involves specifying the data type, attributes and constraints for the data to be stored.

Constructing a database is a process of storing itself on some storage medium like disk or tape that can be handled by the DBMS. Manipulating a data base includes such functions like querying a database to retrieve specific data, updating the database to reflect the changes in the mini world and generate reports from the data.

**DBMS**: A DBMS is a collection of programs that enables us to create maintain and manipulate database. It is hence the general-purpose software system that facilitates the process of defining, constructing and manipulating databases for various applications.

**DATA MODEL:**

Data model is a set of concepts that is used to describe the structure the database.

**HIERARCHICAL DATA MODEL:**

This model represents data hierarchy which has to be maintained.

**DATABASE ABSTRACTION**

A major objective of databases is to provide the user with an abstract view of the data i.e., the system hides the details about the storage details of the database.

**LOW LEVEL ABSTRACTION:**

It defines how exactly the data is stored in the database. It is a detailed view dealing with the actual constructs used and the storage details of the database.

**CONCEPTUAL LEVEL:**

The next level of abstraction defines how exactly the data is stored and a relationship that exists between the data. Implementation of the complex physical storage is hidden from the user.

**VIEW LEVEL:**

This is the highest level of data abstraction and provides the user with facility to view the part of the database. This is to exploit the fact that many users will not need to intricate details of the data storage.

**ABSTRACT:**

In this project we are designing an Olympic database.

Olympic is an international sporting event where many countries participate and players showcase their skills to the world.

It’s a platform for sport, entertainment and good relationship of countries amongst one another.

Olympic database is created in order to record and provide the details of the Olympics.

Olympic database generally consists of the records of the

Players, Countries, Events & Medal Tally. Apart from this we can consider Records, Sponsors and the history of past Olympics.

We here try to implement a real time application using windows forms as GUI and SQL server to hold database. This includes the complete data collection of an Olympic event and options do provide modifications of entities by DBA(Database Administrator) based on new outcome in each Olympic that goes on.

Entities used-

1. Players
2. Country
3. Event
4. Medals
5. Records
6. History
7. Sponsors

**SOFTWARES USED**

**VISUAL STUDIO:**

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

**ARCHITECTURE:**

Visual Studio does not support any programming language, solution or tool intrinsically, instead it allows the plugging of functionality coded as a VSPackage. When installed, the functionality is available as a Service. The IDE provides three services: SVsSolution, which provides the ability to enumerate projects and solutions; SVsUIShell, which provides windowing and UI functionality (including tabs, toolbars and tool windows); and SVsShell, which deals with registration of VSPackages. In addition, the IDE is also responsible for coordinating and enabling communication between services

**SQL SERVER:**

**Microsoft SQL Server** is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft). As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for different workloads (ranging from small applications that store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet at the same time).

**Programming Language**

**CSHARP:**

The correct title of this article is **C# (programming language)**. The substitution or omission of the [#](http://en.wikipedia.org/wiki/Number_sign) sign is because of [technical restrictions](http://en.wikipedia.org/wiki/Wikipedia:Naming_conventions_%28technical_restrictions%29#Forbidden_characters).

**C#** is a [multi-paradigm programming language](http://en.wikipedia.org/wiki/Multi-paradigm_programming_language) encompassing [strong typing](http://en.wikipedia.org/wiki/Strong_typing), [imperative](http://en.wikipedia.org/wiki/Imperative_programming), [declarative](http://en.wikipedia.org/wiki/Declarative_programming), [functional](http://en.wikipedia.org/wiki/Functional_programming), [generic](http://en.wikipedia.org/wiki/Generic_programming) was developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft) within its [.NET](http://en.wikipedia.org/wiki/.NET_Framework) initiative and later approved as a standard by [ISO](http://en.wikipedia.org/wiki/International_Organization_for_Standardization).

C# is one of the programming languages designed for the [Common Language Infrastructure](http://en.wikipedia.org/wiki/Common_Language_Infrastructure).

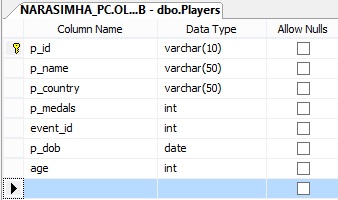
**DESIGN GOALS OF C SHARP**

The ECMA standard lists these design goals for C#

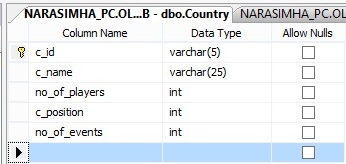
* C# language is intended to be a simple, modern, general-purpose, object-oriented programming language.
* The language, and implementations thereof, should provide support for software engineering principles such as [strong type](http://en.wikipedia.org/wiki/Strong_type) checking, array [bounds checking](http://en.wikipedia.org/wiki/Bounds_checking), detection of attempts to use uninitialized variables, and automatic [garbage collection](http://en.wikipedia.org/wiki/Garbage_collection_%28computer_science%29). Software robustness, durability, and programmer productivity are important.
* The language is intended for use in developing [software components](http://en.wikipedia.org/wiki/Software_components) suitable for deployment in distributed environments.
* Source code portability is very important, as is programmer portability, especially for those programmers already familiar with C and C++.
* Support for [internationalization](http://en.wikipedia.org/wiki/Internationalization_and_localization) is very important.
* C# is intended to be suitable for writing applications for both hosted and [embedded systems](http://en.wikipedia.org/wiki/Embedded_system), ranging from the very large that use sophisticated [operating systems](http://en.wikipedia.org/wiki/Operating_system), down to the very small having dedicated functions.
* Although C# applications are intended to be economical with regard to memory and [processing power](http://en.wikipedia.org/wiki/Processing_power) requirements, the language was not intended to compete directly on performance and size with C or assembly language.

ENTITIES WITH DATATYPES

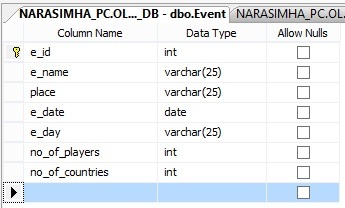
1.PLAYERS



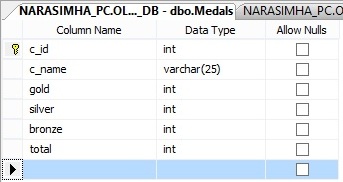
2.COUNTRY



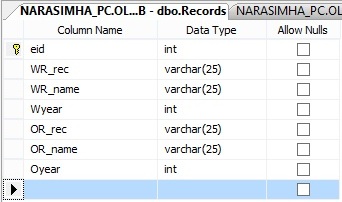
3.EVENTS



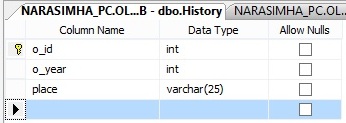
4.MEDALS

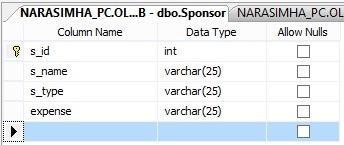


5.RECORDS



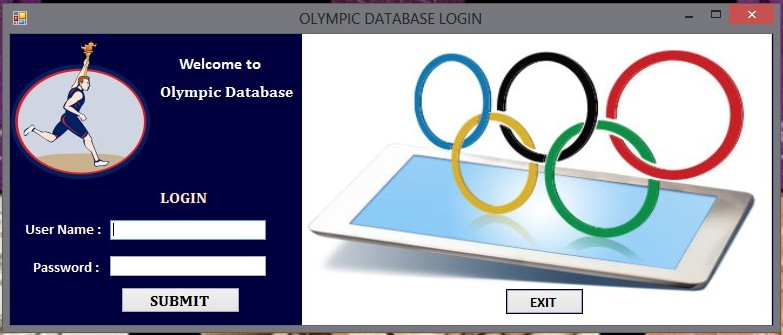
6.HISTORY

7.SPONSORS



WINDOWS FORMS (GUI)

LOGIN



HOME



ENTITIES



OTHER ENTITIES



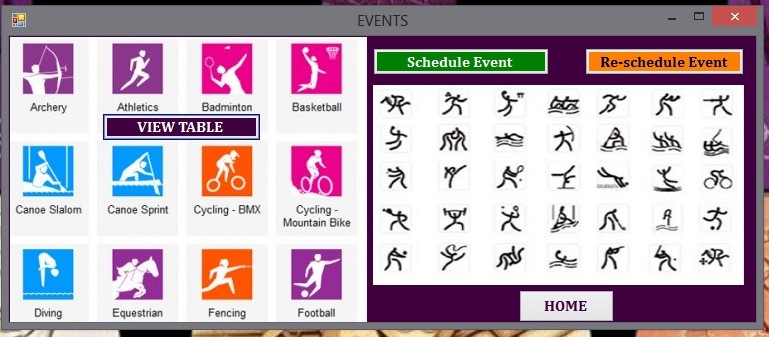
1.PLAYERS



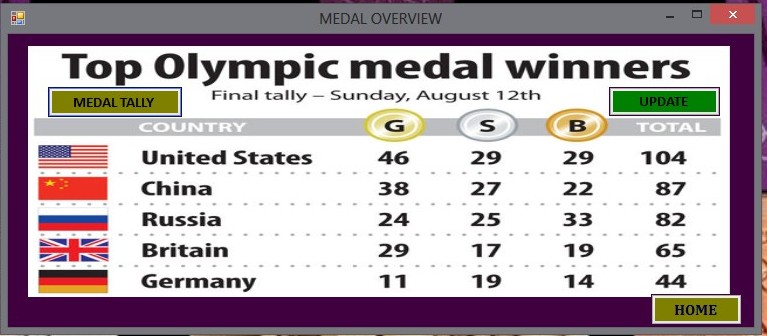
2.COUNTRY



3.EVENTS



4.MEDALS



5.RECORDS



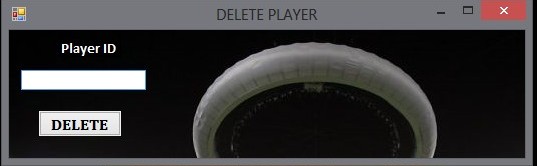
6. HISTORY & SPONSOR



OTHER FORMS

Add player



Delete Player

Update Player



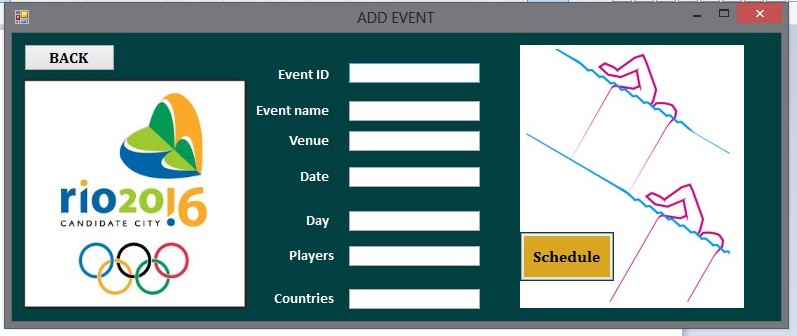
Add Country



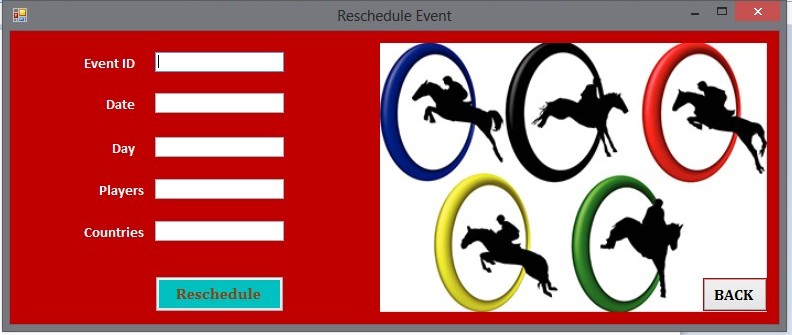
Update Country



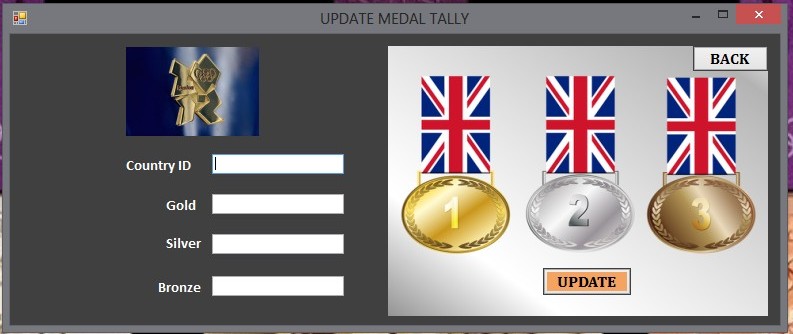
Add Event



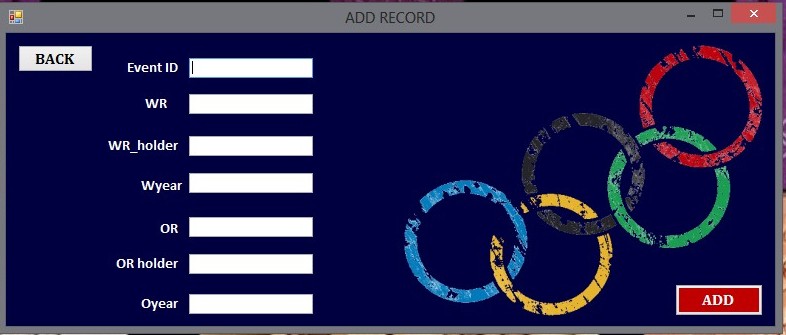
Update Event



Update Medals



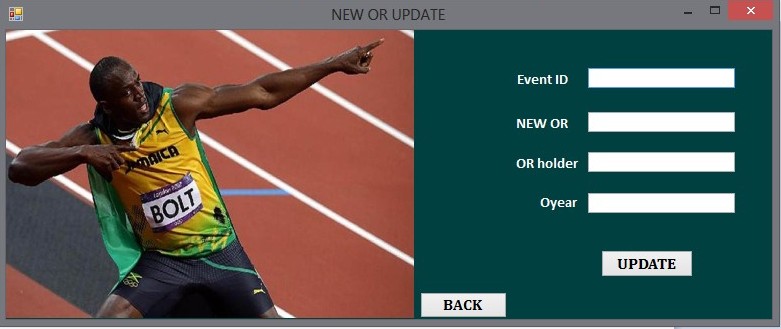
Add Record



New WR



New OR



Database Queries

1.DDL( Database Definition Language)

* CREATE TABLE:

CREATE table <table\_name> (attributes with datatypes & constraints);

eg. CREATE table Players( p\_id varchar(5) ,p\_name varchar(25),

p\_country varchar(25),p\_medals int,event\_id int,p\_dob date,p\_age int,primary key (p\_id));

* INSERT:

INSERT into <table\_name> values (set of values for each attributes);

eg. INSERT into History values (1,2012,”LONDON”);

2.DML(Database Modification Language)

* UPDATE:

UPDATE <table\_name> set attribute\_values WHERE reference\_attribute;

eg. UPDATE Records set WR\_rec = “19.02sec”,WR\_name = “Usain Bolt”,Wyear= 2012 WHERE eid=1;

* DELETE:

DELETE attributes from <table\_name> WHERE reference attribute;

Eg. DELETE ALL from Players WHERE p\_id=9;

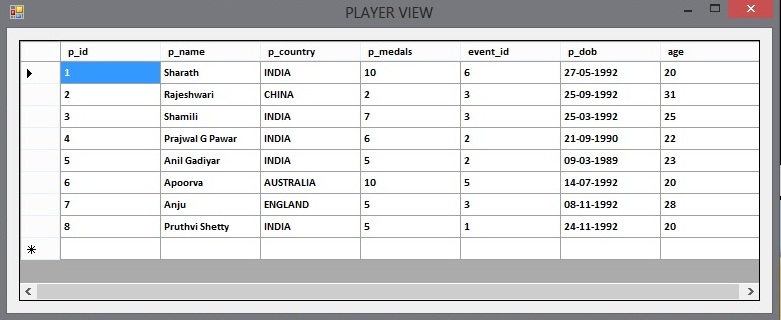
3.VDL(View Definition Language)

SELECT attributes from <table\_name>;

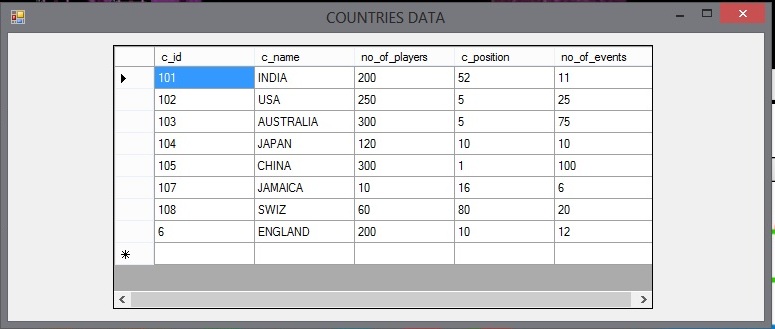
Eg. SELECT \* from Players;

VIEW ENTITIES

1.Players



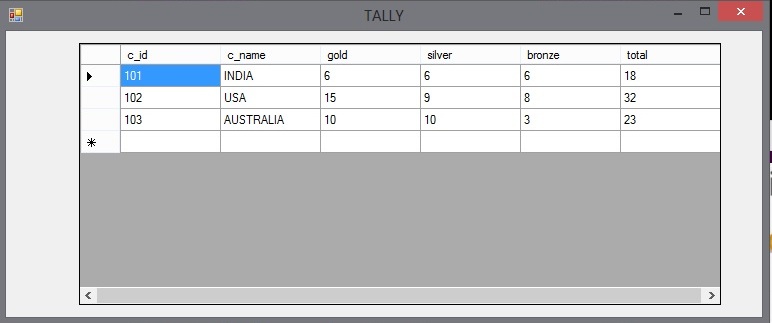
2.Country



3.EVENTS



4.MEDALS



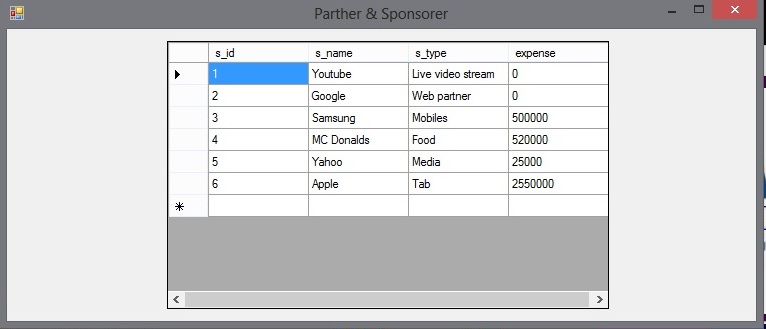
5.RECORDS



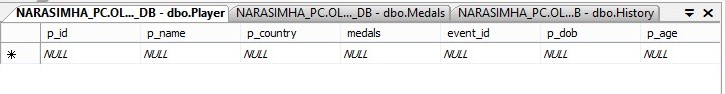
6.HISTORY



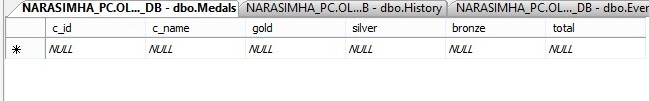
7.SPONSOR

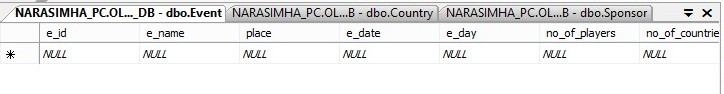


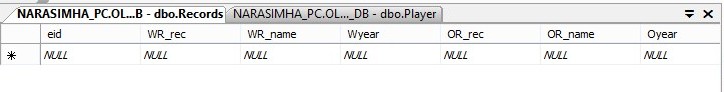
Entity-Relationship flow



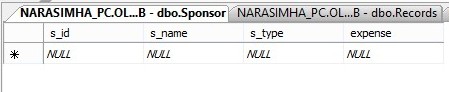












THANKING YOU,

TEAM

NAGARJUN N

SHARATH BABU S

SHAMILY B SHETTY

SANKALP ASTHIK G.K