**.NET面试宝典**

目录

[.NET面试宝典 1](#_Toc10644)

[一、 第1阶段：HTML&c#基础 1](#_Toc3124)

[二、 第2阶段：WEB前端基础 13](#_Toc32074)

[三、 第3阶段：ASP.NET基础 22](#_Toc4759)

[四、 第4阶段：ASP.NET高级 49](#_Toc16055)

[五、 第5阶段：ASP.NET企业级技术 58](#_Toc17714)

.NET面试宝典

1. 第1阶段：HTML&c#基础
   1. 简述 private、 protected、 public、 internal 修饰符的访问权限。

private : 私有成员, 在类的内部才可以访问。

protected : 保护成员，该类内部和继承类中可以访问。

public : 公共成员，完全公开，没有访问限制。

internal: 当前程序集内可以访问。

* 1. ADO.NET中的五个主要对象

**Connection**：主要是开启程序和数据库之间的连接。没有利用连接对象将数据库打开，是无法从数据库中取得数据的。Close和Dispose的区别，Close以后还可以Open，Dispose以后则不能再用。

**Command**：主要可以用来对数据库发出一些指令，例如可以对数据库下达查询、新增、修改、删除数据等指令，以及调用存在数据库中的存储过程等。这个对象是架构在Connection 对象上，也就是Command 对象是透过连接到数据源。

**DataAdapter**：主要是在数据源以及DataSet 之间执行数据传输的工作，它可以透过Command 对象下达命令后，并将取得的数据放入DataSet 对象中。这个对象是架构在Command对象上，并提供了许多配合DataSet 使用的功能。

**DataSet**：这个对象可以视为一个暂存区（Cache），可以把从数据库中所查询到的数据保留起来，甚至可以将整个数据库显示出来，DataSet是放在内存中的。DataSet 的能力不只是可以储存多个Table 而已，还可以透过DataAdapter对象取得一些例如主键等的数据表结构，并可以记录数据表间的关联。DataSet 对象可以说是ADO.NET 中重量级的对象，这个对象架构在DataAdapter对象上，本身不具备和数据源沟通的能力；也就是说我们是将DataAdapter对象当做DataSet 对象以及数据源间传输数据的桥梁。DataSet包含若干DataTable、DataTableTable包含若干DataRow。

**DataReader**：当我们只需要循序的读取数据而不需要其它操作时，可以使用DataReader 对象。DataReader对象只是一次一笔向下循序的读取数据源中的数据，这些数据是存在数据库服务器中的，而不是一次性加载到程序的内存中的，只能（通过游标）读取当前行的数据，而且这些数据是只读的，并不允许作其它的操作。因为DataReader 在读取数据的时候限制了每次只读取一笔，而且只能只读，所以使用起来不但节省资源而且效率很好。使用DataReader 对象除了效率较好之外，因为不用把数据全部传回，故可以降低网络的负载。

ADO.NET 使用Connection 对象来连接数据库，使用Command 或DataAdapter对象来执行SQL语句，并将执行的结果返回给DataReader 或 DataAdapter ,然后再使用取得的DataReader 或DataAdapter 对象操作数据结果。

* 1. 列举ASP.NET 页面之间传递值的几种方式。

1.使用QueryString, 如....?id=1; response. Redirect()....

2.使用Session变量

3.使用Server.Transfer

4.Cookie传值

* 1. 面向对象的三个特性？

**继承性**：就是让一个类型的对象拥有另一个类型的对象的属性的方法。继承后，子类拥有父类的属性和方法。

**封装性**：就是将数据和代码捆绑在一起，防止外界干扰。

**多态性**：就是一个事物拥有不同形式的能力。

* 1. C#中的委托是什么？事件是不是一种委托？事件和委托的关系。

委托可以把一个方法作为参数代入另一个方法。

委托可以理解为指向一个函数的指针。

委托和事件没有可比性，因为委托是类型，事件是对象，下面说的是委托的对象（用委托方式实现的事件）和（标准的event方式实现）事件的区别。事件的内部是用委托实现的。因为对于事件来讲，外部只能“注册自己+=、注销自己-=”，外界不可以注销其他的注册者，外界不可以主动触发事件，因此如果用Delegate就没法进行上面的控制，因此诞生了事件这种语法。事件是用来阉割委托实例的，类比用一个自定义类阉割List。事件只能add、remove自己，不能赋值。事件只能+=、-=，不能= 。加分的补充回答：事件内部就是一个private的委托和add、remove两个方法。

* 1. override与重载的区别

重载是方法的名称相同，参数或参数类型不同，进行多次重载以适应不同的需要。

Override是进行基类中函数的重写，为了适应需要。

* 1. 属性和public字段的区别是什么？调用set方法为一个属性设值，然后用get方法读取出来的值一定是set进去的值吗？

属性可以对设值、取值的过程进行非法值控制，比如年龄禁止设值负数，而字段则不能进行这样的设置。虽然一般情况下get读取的值就是set设置的值，但是可以让get读取的值不是set设置的值的，极端的例子：Public Age{get{return 100;}set{}}。

* 1. 三层架构

通常意义上的三层架构就是将整个业务应用划分为：表现层（UI）、业务逻辑层（BLL）、数据访问层（DAL）。

区分层次的目的即为了“高内聚，低耦合”的思想。

表现层（UI）：通俗讲就是展现给用户的界面，即用户在使用一个系统的时候的所见所得。

业务逻辑层（BLL）：针对具体问题的操作，也可以说是对数据层的操作，对数据业务逻辑处理。

数据访问层（DAL）：该层所做事务直接操作数据库，针对数据的增添、删除、修改、更新、查找等每层之间是一种垂直的关系。

三层结构是N层结构的一种，一般来说，层次之间是向下依赖的，下层代码未确定其接口（契约）前，上层代码是无法开发的，下层代码接口（契约）的变化将使上层的代码一起变化。

优点： 分工明确，条理清晰，易于调试，而且具有可扩展性。

缺点： 增加成本。

* 1. 什么是装箱(boxing)和拆箱(unboxing)？

装箱：从值类型接口转换到引用类型。

拆箱：从引用类型转换到值类型。

object obj = null;//引用类型

obj = 1;//装箱，boxing。把值类型包装为引用类型。

int i1 = (int)obj;//拆箱。Unboxing

* 1. CTS、CLS、CLR分别作何解释？

**CTS**：Common Type System 通用类型系统。Int32、Int16→int、String→string、Boolean→bool。每种语言都定义了自己的类型，.Net通过CTS提供了公共的类型，然后翻译生成对应的.Net类型。

**CLS**：Common Language Specification 通用语言规范。不同语言语法的不同。每种语言都有自己的语法，.Net通过CLS提供了公共的语法，然后不同语言翻译生成对应的.Net语法。

**CLR**：Common Language Runtime 公共语言运行时，就是GC、JIT等这些。有不同的CLR，比如服务器CLR、Linux CLR（Mono）、Silverlight CLR(CoreCLR)。相当于一个发动机，负责执行IL。

* 1. 在.NET中类(class)与结构(struct)的异同？

Class可以被实例化,属于引用类型,是分配在内存的堆上的。类是引用传递的。

Struct属于值类型,是分配在内存的栈上的。结构体是复制传递的。

* 1. 堆和栈的区别？

栈是编译期间就分配好的内存空间，因此你的代码中必须就栈的大小有明确的定义；局部值类型变量、值类型参数等都在栈内存中。

堆是程序运行期间动态分配的内存空间，你可以根据程序的运行情况确定要分配的堆内存的大小。

* 1. 能用foreach遍历访问的对象的要求？

需要实现IEnumerable接口或声明GetEnumerator方法的类型。

* 1. GC是什么? 为什么要有GC?

C/C++中由程序员进行对象的回收像学校食堂中由学生收盘子，.Net中由GC进行垃圾回收像餐馆中店员去回收。

GC是垃圾收集器（Garbage Collection）。程序员不用担心内存管理，因为垃圾收集器会自动进行管理。GC只能处理托管内存资源的释放，对于非托管资源则不能使用GC进行回收，必须由程序员手工回收，一个例子就是FileStream或者SqlConnection需要程序员调用Dispose进行资源的回收。

要请求垃圾收集，可以调用下面的方法：GC.Collect()一般不需要手动调用GC.Collect()。当一个对象没有任何变量指向（不再能使用）的时候就可以被回收了。

基础知识：当没有任何变量指向一个对象的时候对象就可以被回收掉了，但不一定会立即被回收。

* 1. 值类型和引用类型的区别？

1.将一个值类型变量赋给另一个值类型变量时，将复制包含的值。引用类型变量的赋值只复制对对象的引用，而不复制对象本身。

2.值类型不可能派生出新的类型：所有的值类型均隐式派生自 System.ValueType。但与引用类型相同的是，结构也可以实现接口。

3.值类型不可能包含 null 值：然而，可空类型功能允许将 null 赋给值类型。

4.每种值类型均有一个隐式的默认构造函数来初始化该类型的默认值。

* 1. C#中的接口和类有什么异同？

**不同点**：

不能直接实例化接口。

接口不包含方法的实现。

接口可以多继承，类只能单继承。

类定义可在不同的源文件之间进行拆分。

**相同点**：

接口、类和结构都可以从多个接口继承。

接口类似于抽象基类：继承接口的任何非抽象类型都必须实现接口的所有成员。

接口和类都可以包含事件、索引器、方法和属性。

基础知识：接口只能定义方法（只能定义行为，不能定义实现也就是字段），因为事件、索引器、属性本质上都是方法，所以接口中也可以定义事件、索引器、属性。

* 1. abstract class和interface有什么区别？

**相同点：**

都不能被直接实例化，都可以通过继承实现其抽象方法。

**不同点：**

接口支持多继承；抽象类不能实现多继承。

接口只能定义行为；抽象类既可以定义行为，还可能提供实现。

接口只包含方法(Method)、属性(Property)、索引器(Index)、事件(Event)的签名，但不能定义字段和包含实现的方法；

抽象类可以定义字段、属性、包含有实现的方法。

接口可以作用于值类型(Struct)和引用类型(Class)；抽象类只能作用于引用类型。例如，Struct就可以继承接口，而不能继承类。

* 1. 是否可以继承String类？

String类是sealed类故不可以继承。

* 1. using关键字有什么用？什么是IDisposable？

using可以声明namespace的引入，还可以实现非托管资源的释放，实现了IDisposiable的类在using中创建，using结束后会自动调用该对象的Dispose方法，释放资源。加分的补充回答：using其实等价于try……finally，用起来更方便。

* 1. string str = null 与 string str = “”说明其中的区别？

string str = null 是不给他分配内存空间,而string str = “” 给它分配长度为空字符串的内存空间。 string str = null没有string对象，string str = “”有一个字符串对象。

* 1. 什么是code-Behind技术？

就是代码隐藏（代码后置），在ASP.NET中通过ASPX页面指向CS文件的方法实现显示逻辑和处理逻辑的分离，这样有助于web应用程序的创建。比如分工，美工和编程的可以个干各的，不用再像以前asp那样都代码和html代码混在一起，难以维护。code-Behind是基于部分类技术实现的，在我的项目的三层代码生成器中用到了部分类。

* 1. StringBuilder 和 String 的区别？

String 在进行运算时（如赋值、拼接等）会产生一个新的实例，而 StringBuilder 则不会。所以在大量字符串拼接或频繁对某一字符串进行操作时最好使用 StringBuilder，不要使用 String。

如果要操作一个不断增长的字符串，尽量不用String类，改用StringBuilder类。两个类的工作原理不同：String类是一种传统的修改字符串的方式，它确实可以完成把一个字符串添加到另一个字符串上的工作没错,但是在.NET框架下，这个操作实在是划不来。因为系统先是把两个字符串写入内存，接着删除原来的String对象，然后创建一个String对象，并读取内存中的数据赋给该对象。这一来二去的，耗了不少时间。而使用System.Text命名空间下面的StringBuilder类就不是这样了，它提供的Append方法，能够在已有对象的原地进行字符串的修改，简单而且直接。当然，一般情况下觉察不到这二者效率的差异，但如果你要对某个字符串进行大量的添加操作，那么StringBuilder类所耗费的时间和String类简直不是一个数量级的。

* 1. 不是说字符串是不可变的吗？string s="abc";s="123"不就是变了吗？

String是不可变的在这段代码中，s原先指向一个String对象，内容是 "abc"，然后我们将s指向"123"，那么s所指向的那个对象是否发生了改变呢？答案是没有。这时，s不指向原来那个对象了，而指向了另一个 String对象，内容为"123"，原来那个对象还存在于内存之中，只是s这个引用变量不再指向它了。

* 1. 是否可以从一个static方法内部发出对非static方法的调用？

不可以。因为非static方法是要与对象关联在一起的，必须创建一个对象后，才可以在该对象上进行方法调用，而static方法调用时不需要创建对象，可以直接调用。也就是说，当一个static方法被调用时，可能还没有创建任何实例对象，如果从一个static方法中发出对非static方法的调用，那个非static方法是关联到哪个对象上的呢？这个逻辑无法成立，所以，一个static方法内部不能发出对非static方法的调用。

* 1. 说出一些常用的类、接口，请各举5个？

要让人家感觉你对.Net开发很熟，所以，不能仅仅只列谁都能想到的那些东西，要多列你在做项目中涉及的那些东西。就写你最近写的那些程序中涉及的那些类。

常用的类：StreamReader、WebClient、Dictionary<K,V>、StringBuilder、SqlConnection、FileStream、File、Regex、List<T>

常用的接口：IDisposable、IEnumerable、IDbConnection、IComparable、ICollection、IList、IDictionary

* 1. 编写一个单例（Singleton）类？

把构造函数设置为private，设置一个public、static的对象实例

public class FileManager

{

private FileManager(){}

public readonly static FileManager Instance = new FileManager();

}

* 1. .Net、ASP.Net、C#、VisualStudio之间的关系是什么？

.Net一般指的是.Net Framework，提供了基础的.Net类，这些类可以被任何一种.Net编程语言调用，.Net Framework还提供了CLR、JIT、GC等基础功能。

ASP.Net是.Net中用来进行Web开发的一种技术，ASP.Net的页面部分写在aspx 文件中，逻辑代码通常通过Code-behind的方式用C#、VB.Net等支持.Net的语言编写。

C#是使用最广泛的支持.Net的编程语言。除了C#还有VB.Net、IronPython等。

VisualStudio是微软提供的用来进行.Net开发的集成开发环境（IDE），使用VisualStudio可以简化很多工作，不用程序员直接调用csc.exe等命令行进行程序的编译，而且VisualStudio提供了代码自动完成、代码高亮等功能方便开发。除了VisualStudio，还有SharpDevelop、MonoDevelop等免费、开源的IDE，VisualStudio Express版这个免费版本。

* 1. 以下两种异常处理方式有什么区别？try {} catch(){throw;} try{}catch(Exception ex){Throw ex;}

前者catch无参数，可以捕获所有异常，只有throw关键字，表示抛出当前catch语句捕获的异常。

后者catch有参数，可以捕获所有以Exception类派生的异常，throw 变量名；抛出的是Exception异常或Exception派生的类型

* 1. 用接口或父类作为输入的参数有什么好处？返回值如果是接口是为了什么？

接口或父类作为参数，所有继承了此接口或父类的类都可以使用，即你给此方法传参数的时候，可以把继承了这个接口或父类的类的实例传给这个方法。可以一次性传入多种字段、方法等而不用定义多个形参。

接口作为返回值，返回的是实现接口的对象。

* 1. C#和JS的闭包？

闭包的概念是内层的函数可以引用包含在它外层的函数的变量，即使外层函数的执行已经终止。但该变量提供的值并非变量创建时的值，而是在父函数范围内的最终值。

C#闭包可理解为跨作用域访问函数内变量，那么如何避免闭包陷阱呢？C#中普遍的做法是，将匿名函数引用的变量用一个临时变量保存下来，然后在匿名函数中使用临时变量。JS闭包，在js中的我的理解就是函数嵌套函数，当内部函数在定义它的作用域的外部被引用时,就创建了该内部函数的闭包 ,如果内部函数引用了位于外部函数的变量,当外部函数调用完毕后,这些变量在内存不会被释放,因为闭包需要它们。

* 1. Sleep()和wait()有什么区别？

1. sleep是线程类Thread 的方法，它是使当前线程暂时睡眠，可以放在任何位置。

而wait，它是使当前线程暂时放弃对象的使用权进行等待，必须放在同步方法或同步块里。

2.Sleep使用的时候，线程并不会放弃对象的使用权，即不会释放对象锁，所以在同步方法或同步块中使用sleep，一个线程访问时，其他的线程也是无法访问的。

而wait是会释放对象锁的，就是当前线程放弃对象的使用权，让其他的线程可以访问。

3.线程执行wait方法时，需要其他线程调用Monitor.Pulse()或者Monitor.PulseAll()进行唤醒或者说是通知等待的队列。

而sleep只是暂时休眠一定时间，时间到了之后，自动恢复运行，不需另外的线程唤醒。

* 1. 在c#中using和new这两个关键字有什么意义，请写出你所知道的意义？

using 引入名称空间或者使用非托管资源，使用完对象后自动执行实现了IDisposable接口的类的Dispose方法。

new 新建实例或者隐藏父类方法。

* 1. string str=null与string str="" 有什么区别？

string str=null 把这个引用指向了一个null，没有地址没有值的地方，即没分配内存空间

string str="" 把这个引用指向了一个地址，地址里面存的是空的字符，即占用了内存空间

* 1. 不用中间变量交换两个变量？

int i = 500;

int j = int.MaxValue - 10;

//int i = 10;

//int j = 20;

Console.WriteLine("i={0},j={1}", i, j);

i = i + j;//i=30

j = i - j;//j=10;

i = i - j;//i=20;

Console.WriteLine("i={0},j={1}",i,j);

另外一个解决方案：位运算。

int a=5;

int b=6;

a=a^b;

b=b^a; //b^a相当于 b^a^b 也就是 b^a^b的值就是a了, 下边相同

a=a^b;

* 1. 一个数组：1,1,2,3,5,8,13,21...+m，求第30位数是多少？

写递归要确定两个：递归的终止条件；递归表达式。

解答：总结递归规律：F(n)=F(n-2)+F(n-1) Fibonacci数列

static int F(int n)

{

if (n == 1)

{

return 1;

}

if (n == 2)

{

return 1;

}

return F(n - 2) + F(n - 1);

}

非递归(有bug吗？)，递归算法的缺点：测试大数据

int n = Convert.ToInt32(Console.ReadLine());

if(n<=0)

{Console.WRiteLine("必须大于0");return;}

if(n==1) //时刻注意边界值！！！

{

Console.WriteLine("1");

return;

}

int[] data =new int[n];

data[0] = 1;

data[1] = 1;

for(int i=2;i<n;i++)

{

data[i] = data[i-1]+data[i-2];

}

Console.WriteLine(data[n-1]);

上面程序的时间复杂度为O(n)，空间复杂度为O(n)

参考：

算法复杂度：时间复杂度：算法运行需要消耗的时间的数量级、空间复杂度：算法运行需要消耗的内存的数量级。

消耗的时间或者内存随着问题规模的扩大而成正比增加，就是O(n)。

消耗的时间或者内存随着问题规模的扩大而不变，就是O(1)。

消耗的时间或者内存随着问题规模的扩大而n\*n增加，就是O(n\*n)

算法复杂度只考虑最差情况（从一个数组中找出第一个大于10的数，时间复杂度为O(n)），并且算法复杂度忽略常量和低阶。把数组数一遍和数两遍的时间复杂度都是O(n)。把长度为n的数组数n/2遍的时间复杂度还是O(n\*n)。

* 1. 冒泡排序

for (int j = 0; j < nums.Length - 1; j++)

{

for (int i = 0; i < nums.Length - 1 -j; i++)

{

if (nums[i] > nums[i + 1])

{

int temp = nums[i];

nums[i] = nums[i + 1];

nums[i + 1] = temp;

}

}

}

* 1. 求以下表达式的值，写出您想到的一种或几种实现方法：1-2+3-4+……+m

int Num = this.TextBox1.Text.ToString() ;

int Sum = 0 ;

for (int i = 0 ; i < Num + 1 ; i++)

{

if((i%2) == 1)

{

Sum += i ;

}

else

{

Sum = Sum - I ;

}

}

System.Console.WriteLine(Sum.ToString());

System.Console.ReadLine() ;

1. 第2阶段：WEB前端基础
   1. JavaScript中的“=、==、===”区别？

**=**：即赋值运算；

**==**：判断两个变量是否相同，仅限于值，如果值相同而类型不同，那么JavaScript引擎会在内部做类型转换；

**===**：判断两个变量是否相同，无论是值还是类型，如果类型不同而值相同，也会返回false，而引擎不会在内部进行转换。

* 1. JavaScript中的“undefined、null”区别？

通俗地讲，undefined出现的原因是JavaScript引擎不知道这是个什么东西，而对于null，JavaScript引擎识别了它，但是没有被分配内存空间。

undefined的类型就是undefined，而null的类型是object。

* 1. 简述javascript的作用域和闭包？

js变量的作用域是指：函数内定义的局部变量只在此函数内有效，而全局变量可以全局有效。

闭包的作用就在于能够改变局部变量的作用域，将值保存下来，但是如果使用不当会造成无法回收变量，引起性能问题，甚至崩溃。

* 1. 对web标准以及w3c的理解和认识？

标签闭合，标签小写，不乱嵌套，提高搜索机器人搜索几率。使用外链css和js脚本，结构行为表现分离，内容能被更多广泛的设备所访问,更少的代码和组件，容易维护，改版方便，不需要变动页面内容。

* 1. Html和xhtml有什么区别？

html是一种基本的web网页设计语言，xhtml是一个基于XML的置标语言。

最主要的不同:

XHTML元素必须正确的被嵌套，元素必须关闭，标签必须小写，必须有根元素。

* 1. 严格模式和混杂模式的区分，以及如何触发这2种模式？

严格模式就是浏览器根据web标准去解析页面，是一种要求严格的DTD，不允许使用任何表现层的语法。

混杂模式是一种向后兼容的解析方法。

触发标准模式或者说严格模式很简单，就是Html前申明正确的DTD，出发混杂模式可以在html文档开始不声明DTD，或者在DOCTYPE前加入XML声明。

* 1. 行内元素有哪些?块级元素有哪些？CSS盒模型？

行内元素: br span input a

块级元素IV P H1 H2 FORM ul

CSS盒模型内容:padding border margin

* 1. css引入的方式有哪些?link和@import的区别？

引入方式3种：行内添加定义style属性值，页面头部内内嵌调用和外链调用，

区别：

1.link是xhtml标签，除了加载css外，还可以定义RSS等其他事务，@import只能加载CSS

2.link引用CSS时候，页面载入的时候同时加载，@import需要页面网页完全载入后加载

3.link是XHTML标签，没有兼容问题，@import是在CSS2.1提出的，低版本的浏览器不支持。

4.link支持使用javascript控制DOM去改变样式，但是@import不支持。

* 1. CSS选择符有哪些?优先级算法如何计算?内联和Important哪个优先级高？

CSS选择符有类选择符，属性选择符，ID选择符，优先级算法是基于特殊性值进行计算的。分别如下：ID属性 0.1.0.0

类属性选择器，属性选择器，伪类 0.0.1.0 元素选择符，伪元素选择符 0.0.0.1

通配选择器对特殊性没有任何贡献值。

important优先级高

* 1. 前端页面有哪三层构成，分别是什么，作用是什么？

结构层：html 表示层:css 行为层：js

* 1. 页面的哪些浏览器你测试过，分别内核是什么？

IE: trident内核

Firefox：gecko内核

Safari:webkit内核

Opera:以前是presto内核，Opera现已改用Google Chrome的Blink内核

Chrome:Blink(基于webkit，Google与Opera Software共同开发)

* 1. img标签上的title和alt属性区别是什么？

alt是当图片不能正常显示的时候，用文字代替

title该属性提供信息

* 1. 你如何对网站的文件和资源进行优化？

文件合并

文件最小化/文件压缩

使用CDN托管（CDN的全称是Content Delivery Network，即内容分发网络。其基本思路是尽可能避开互联网上有可能影响数据传输速度和稳定性的瓶颈和环节，使内容传输的更快、更稳定。）

缓存的使用

* 1. 什么是语义化的HTML？

直观的认识标签

对于搜索引擎的抓取有好处

* 1. 清除浮动的几种方式，各自的优缺点？

使用空标签清除浮动 clear:both（理论上能清楚任何标签，增加无意义的标签）

使用overflow:auto（空标签元素清除浮动而不得不增加无意代码的弊端，使用zoom:1用于兼容IE）

使用afert伪元素清除浮动(用于非IE浏览器)

* 1. javascript的typeof返回哪些数据类型？

Object number function boolean underfind

* 1. 例举3种强制类型转换和2种隐式类型转换？

强制（parseInt,parseFloat,number）

隐式（== – ===）

* 1. split() join() 的区别？

前者是切割成数组的形式，后者是将数组转换成字符串

* 1. 数组方法pop() push() unshift() shift()

Push()尾部添加 pop()尾部删除

Unshift()头部添加 shift()头部删除

* 1. 闭包是什么，有什么特性，对页面有什么影响？

闭包就是能够读取其他函数内部变量的函数。

详情查看这篇博文：<http://blog.csdn.net/gaoshanwudi/article/details/7355794>

* 1. 解释jsonp的原理，以及为什么不是真正的ajax？

动态创建script标签，回调函数

Ajax是页面无刷新请求数据操作

* 1. javascript的本地对象，内置对象和宿主对象？

本地对象为array obj regexp等可以new实例化

内置对象为gload Math 等不可以实例化的

宿主为浏览器自带的document,window 等

* 1. document load 和document ready的区别？

Document.onload 是在结构和样式加载完才执行js

Document.ready原生种没有这个方法，jquery中有 $().ready(function)

* 1. javascript的同源策略？

一段脚本只能读取来自于同一来源的窗口和文档的属性，这里的同一来源指的是主机名、协议和端口号的组合

* 1. 编写一个数组去重的方法

function oSort(arr)

{

var result ={};

var newArr=[];

for(var i=0;i<arr.length;i++)

{

if(!result[arr])

{

newArr.push(arr)

result[arr]=1

}

}

return newArr

}

* 1. jQuery中有哪几种类型的选择器？

从我自己的角度来讲，可以有3种类型的选择器，如下：

基本选择器：直接根据id、css类名、元素名返回匹配的dom元素。

层次选择器：也叫做路径选择器，可以根据路径层次来选择相应的DOM元素。

过滤选择器：在前面的基础上过滤相关条件，得到匹配的dom元素。

* 1. 请使用jQuery将页面上的所有元素边框设置为2px宽的虚线？

这正是jQuery选择器上场的时候了，代码如下：

<script language="javascript" type="text/javascript">

$("\*").css("border", "2px dotted red");

</script>

* 1. 当CDN上的jQuery文件不可用时，该怎么办？

为了节省带宽和脚本引用的稳定性，我们会使用CDN上的jQuery文件，例如google的jquery cdn服务。但是如果这些CDN上的jQuery服务不可用，我们还可以通过以下代码来切换到本地服务器的jQuery版本：

<script type="text/javascript" language="Javascript" src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.4.1.min.js "></script>

<script type='text/javascript'>//<![CDATA[

if (typeof jQuery == 'undefined') {

document.write(unescape("%3Cscript src='/Script/jquery-1.4.1.min.js' type='text/javascript' %3E%3C/script%3E"));

}//]]>

</script>

* 1. 怎样用jQuery编码和解码URL？

在jQuery中，我们可以使用以下方法实现URL的编码和解码。

encodeURIComponent(url) and decodeURIComponent(url)

* 1. 如何用jQuery禁用浏览器的前进后退按钮？

实现代码如下：

<script type="text/javascript" language="javascript">

$(document).ready(function() {

window.history.forward(1);

//OR

window.history.forward(-1);

});

</script>

* 1. 你为什么要使用jquery？

因为jQuery是轻量级的框架，大小不到30kb,它有强大的选择器，出色的DOM操作的封装，有可靠的事件处理机制(jQuery在处理事件绑定的时候相当的可靠)，完善的ajax(它的ajax封装的非常的好，不需要考虑复杂浏览器的兼容性和XMLHttpRequest对象的创建和使用的问题。) 出色的浏览器的兼容性。 而且支持链式操作，隐式迭代。行为层和结构层的分离，还支持丰富的插件，jquery的文档也非常的丰富。

* 1. jquery对象和dom对象是怎样转换的？

jquery转DOM对象:jQuery 对象是一个数组对象，可以通过[index]的丰富得到相应的DOM对象还可以通过get[index]去得到相应的DOM对象。DOM对象转jQuery对象:$(DOM对象)

* 1. 你是如何使用jquery中的ajax的？

如果是一些常规的ajax程序的话，使用load(),$.get(),$.post(),就可以搞定了，一般我会使用的是$.post() 方法。如果需要设定beforeSend(提交前回调函数),error(失败后处理),success(成功后处理)及complete(请求完成后处理)回调函数等，这个时候我会使用$.ajax()

* 1. 你觉得jquery中的ajax好用吗，为什么？

好用的。 因为jQuery提供了一些日常开发中的快捷操作，例 load，ajax，get，post等等，所以使用jQuery开发ajax将变得极其简单，我们就可以集中精力在业务和用户的体验上，不需要去理会那些繁琐的XMLHttpRequest对象了。

* 1. jquery中$.get()提交和$.post()提交有区别吗？

$.get() 方法使用GET方法来进行异步请求的。$.post() 方法使用POST方法来进行异步请求的。

get请求会将参数跟在URL后进行传递，而POST请求则是作为HTTP消息的实体内容发送给Web服务器的，这种传递是对用户不可见的。

get方式传输的数据大小不能超过2KB 而POST要大的多

GET 方式请求的数据会被浏览器缓存起来，因此有安全问题。

* 1. 在jquery中你是如何去操作样式的？

addClass() 来追加样式 ，removeClass() 来删除样式，toggle() 来切换样式

* 1. 你使用过jquery中的动画吗，是怎样用的？

使用过。

hide() 和 show() 同时修改多个样式属性。像高度，宽度，不透明度。 fadeIn() 和fadeOut() fadeTo() 只改变不透明度

slideUp() 和 slideDown() slideToggle() 只改变高度

animate() 属于自定义动画的方法

* 1. jQuery是如何处理缓存的？

要处理缓存就是禁用缓存。

通过$.post() 方法来获取数据，那么默认就是禁用缓存的。

通过$.get()方法 来获取数据，可以通过设置时间戳来避免缓存。可以在URL后面加上+(+new Date)例 $.get('ajax.xml?'+(+new Date),function () { //内容 }); 3 通过$.ajax 方法来获取数据，只要设置cache:false即可。

* 1. $.getScript()方法 和 $.getJson() 方法有什么区别？

$.getScript() 方法可以直接加载.js文件，并且不需要对javascript文件进行处理，javascript文件会自动执行。

$.getJson() 是用于加载JSON 文件的 ，用法和$.getScript()

* 1. 你使用过哪些数据格式，它们各有什么特点？

HTML格式 ,，JSON格式，javascript格式，XML格式

HTML片段提供外部数据一般来说是最简单的。

如果数据需要重用，而且其他应用程序也可能一次受到影响，那么在性能和文件大小方面具有优势的JSON通常是不错的选择。

而当远程应用程序未知时，XML则能够为良好的互操作性提供最可靠的保证。

* 1. jQuery 能做什么？

获取页面的元素

修改页面的外观

改变页面大的内容

响应用户的页面操作

为页面添加动态效果

无需刷新页面，即可以从服务器获取信息

简化常见的javascript任务

* 1. 在ajax中data主要有几种方式？

三种，html拼接的，json数组，form表单经serialize()序列化的。

* 1. 你知道jQuery中的事件冒泡吗，它是怎么执行的，如何来停止冒泡事件？

知道，事件冒泡是从里面的往外面开始触发。在jQuery中提供了stopPropagation()方法可以停止冒泡。

1. 第3阶段：ASP.NET基础
   1. 请解释ASP。NET中的web页面与其隐藏类之间的关系？

一个ASP.NET页面一般都对应一个隐藏类，一般都在ASP.NET页面的声明中指定了隐藏类例如一个页面Tst1.aspx的页面声明如下：

<%@ Page language="c#" Codebehind="Tst1.aspx.cs" AutoEventWireup="false" Inherits="T1.Tst1" %>

Codebehind="Tst1.aspx.cs" 表明经编译此页面时使用哪一个代码文件。

Inherits="T1.Tst1" 表用运行时使用哪一个隐藏类。

aspx页面会编译生成一个类，这个类从隐藏类继承。

* 1. 在什么情况下会用到虚方法？它与接口有什么不同？

子类重新定义父类的某一个方法时，必须把父类的方法定义为virtual

在定义接口中不能有方法体，虚方法可以。

实现时，子类可以不重新定义虚方法，但如果一个类继承接口，那必须实现这个接口。

* 1. DataReader和DataSet的异同？

DataReader使用时始终占用SqlConnection，在线操作数据库

每次只在内存中加载一条数据,所以占用的内存是很小的是只进的、只读的

DataSet则是将数据一次性加载在内存中，抛弃数据库连接，读取完毕即放弃数据库连接(非连接模式)

DataSet将数据全部加载在内存中.所以比较消耗内存，但是确比DataReader要灵活，可以动态的添加行、列、数据，对数据库进行回传更新操作(动态操作读入到内存的数据)。

* 1. Server.Transfer和Response.Redirect的区别是什么？

Server.Transfer仅是服务器中控制权的转向，在客户端浏览器地址栏中不会显示出转向后的地址；Response.Redirect则是完全的跳转，浏览器将会得到跳转的地址，并重新发送请求链接。这样，从浏览器的地址栏中可以看到跳转后的链接地址。

Server.Transfer是服务器请求资源，服务器直接访问目标地址的URL，把那个URL的响应内容读取过来，然后把这些内容再发给浏览器，浏览器根本不知道服务器发送的内容是从哪儿来的，所以它的地址栏中还是原来的地址。 这个过程中浏览器和Web服务器之间经过了一次交互。

Response.Redirect就是服务端根据逻辑,发送一个状态码,告诉浏览器重新去请求那个地址，一般来说浏览器会用刚才请求的所有参数重新请求。这个过程中浏览器和Web服务器之间经过了两次交互。

* 1. 什么是sql注入？如何避免sql注入？

用户根据系统的程序构造非法的参数从而导致程序执行不是程序员期望的恶意SQL语句。使用参数化的SQL就可以避免SQL注入。

* 1. 数据库三范式是什么？

第一范式：字段不能有冗余信息，所有字段都是必不可少的。

第二范式：满足第一范式并且表必须有主键。

第三范式：满足第二范式并且表引用其他的表必须通过主键引用。

* 1. post、get的区别？

get的参数会显示在浏览器地址栏中，而post的参数不会显示在浏览器地址栏中；

使用post提交的页面在点击【刷新】按钮的时候浏览器一般会提示“是否重新提交”，而get则不会；

用get的页面可以被搜索引擎抓取，而用post的则不可以；

用post可以提交的数据量非常大，而用get可以提交的数据量则非常小(2k)，受限于网页地址的长度。

用post可以进行文件的提交，而用get则不可以。

* 1. Session,ViewState,Application,cookie的区别？

**Session**:用于保持状态的基于 Web 服务器的方法。Session 允许通过将对象存储在Web 服务器的内存中在整个用户会话过程中保持任何对象。主要用于保持代码隐藏类中对象的状态。为每个用户创建的,用于存储单个用户,因为他是相对每个用户的.所以可能来取得在线人数等。

**ViewState**:主要用于保持 Web 页上控件的状态。当 Web 页上的控件被绑定到代码隐藏类中的对象。

**Application** 用于存储所有用户都可视的信息.所以它存储的是要让所有用户共享的一些信息.如总访问数等Cache，页面缓存。

**Cookie**:通常我们都把它放在客户端,也可以存储在服务器端。主要用它存储用户的个性设制,和登录信息。

* 1. 写出一条Sql语句：取出表A中第31到第40记录（SQLServer,以自动增长的ID作为主键,注意：ID可能不是连续的。

解1: select top 10 \* from A where id not in (select top 30 id from A)

演变步骤：

1）select top 30 id from T\_FilterWords--取前条

2）select \* from T\_FilterWords

where id not in (select top 30 id from T\_FilterWords)--取id不等于前三十条的

--也就是把前条排除在外

3）select top 10 \* from T\_FilterWords

where id not in (select top 30 id from T\_FilterWords)

--取把前条排除在外的前条，也就是-40条

解2: select top 10 \* from A where id > (select max(id) from (select top 30 id from A )as A)

解答3：用ROW\_NUMBER实现

* 1. 横表、纵表转换

**纵表结构 TableA**

|  |  |  |
| --- | --- | --- |
| **Name** | **Course** | **Grade** |
| **张三** | **语文** | **75** |
| **张三** | **数学** | **80** |
| **张三** | **英语** | **90** |
| **李四** | **语文** | **95** |
| **李四** | **数学** | **55** |

**横表结构 TableB**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **语文** | **数学** | **英语** |
| **张三** | **75** | **80** | **90** |
| **李四** | **95** | **55** | **0** |

先理解：

select Name，

(case Course when ‘语文‘ then Grade else 0 end) as 语文，

(case Course when ‘数学‘ then Grade else 0 end) as 数学，

(case Course when ‘英语‘ then Grade else 0 end) as 英语

from TableA

然后理解标准答案：

select Name，

sum(case Course when ‘语文‘ then Grade else 0 end) as 语文，

sum(case Course when ‘数学‘ then Grade else 0 end) as 数学，

sum(case Course when ‘英语‘ then Grade else 0 end) as 英语

from TableA

group by Name

**横表转纵表的"SQL"示例**

横表结构: TEST\_H2Z

ID 姓名 语文 数学 英语

1 张三 80 90 70

2 李四 90 85 95

3 王五 88 75 90

转换后的表结构:

ID 姓名 科目 成绩

1 张三 语文 80

2 张三 数学 90

3 张三 英语 70

4 李四 语文 90

5 李四 数学 80

6 李四 英语 99

7 王五 语文 85

8 王五 数学 96

9 王五 英语 88

横表转纵表SQL示例:  
SELECT   姓名,'语文'   AS     科目,语文   AS   成绩   FROM   TEST\_H2Z   UNION   ALL   
SELECT   姓名,'数学'   AS     科目,数学   AS   成绩   FROM   TEST\_H2Z   UNION   ALL   
SELECT   姓名,'英语'   AS     科目,英语   AS   成绩   FROM   TEST\_H2Z  
ORDER BY 姓名,科目 DESC;

* 1. 删除姓名、年龄重复的记录

**Id name age salary**

**1 yzk 80 1000**

**2 yzk 80 2000**

**3 tom 20 20000**

**4 tom 20 20000**

**5 im 20 20000**

//取得不重复的数据

select \* from Persons

where Id in

(

SELECT MAX(Id) AS Expr1

FROM Persons

GROUP BY Name, Age

)

根据姓名、年龄分组，取出每组的Id最大值，然后将Id最大值之外的排除。

删除重复的数据：

delete from Persons

where Id not in

(

SELECT MAX(Id) AS Expr1

FROM Persons

GROUP BY Name, Age

)

**表一：student\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **学号** | **姓名** | **性别** | **出生年月** | **家庭住址** | **备注** |
| **0001** | **张三** | **男** | **1981-8-9** | **北京** | **NULL** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**表二：curriculum**

|  |  |  |
| --- | --- | --- |
| **课程编号** | **课程名称** | **学分** |
| **0001** | **计算机基础** | **2** |
| **0002** | **C语言** | **2** |

**表三：grade**

|  |  |  |
| --- | --- | --- |
| **学号** | **课程编号** | **分数** |
| **0001** | **0001** | **80** |
| **0001** | **0002** | **90** |

**题目：**

**条件查询：**

1. 在GRADE表中查找80-90份的学生学号和分数

select 学号,分数 from grade where 分数 between 80 and 90

1. 在GRADE 表中查找课程编号为003学生的平均分

select avg(分数) from grade where 课程编号='003'

1. 在GRADE 表中查询学习各门课程的人数

Select课程编号,count(学号) as 人数from grade group by 课程编号

1. 查询所有姓张的学生的学号和姓名

select 姓名,学号 from student\_info where 姓名 like '张%'

**嵌套查询：**

1. 查询和学号’0001’的这位同学性别相同的所有同学的姓名和出生年月

select 姓名,出生年月 from student\_info where 性别 in(select 性别 from student\_info where sno='0001')

1. 查询所有选修课程编号为0002 和0003的学生的学号、姓名和性别

select 学号,姓名,性别 from student\_info where 学号 in(select 学号 from grade where 课程编号='0002' and 学号 in(select 学号 from grade where 课程编号='0001'))

1. 查询出学号为0001的学生的分数比0002号学生最低分高的课程编号的课程编号和分数

select 课程编号, 分数 from grade where 学号='0001' and 分数>(select min(分数) from grade where 学号='0002')

**多表查询：**

1. 查询分数在80-90分的学生的学号、姓名、分数

select student\_info.学号,student\_info.姓名,grade.分数 from student\_info,grade where grade.分数 between 80 and 90

1. 查询学习了’C语言’课程的学生学号、姓名和分数

select student\_info.学号,student\_info.姓名,grade.成绩from student\_info,grade,curriculum where student\_info.学号=grade.学号and grade.课程号=curriculum.课程号and curriculum.课程名='C语言'

1. 查询所有学生的总成绩，要求列出学号、姓名、总成绩，没有选课的学生总成绩为空。

select grade.学号,student\_info.姓名,sum(grade.成绩) as 总成绩from student\_info,grade where grade.学号=student\_info.学号group by grade.学号,student\_info.姓名

* 1. 活期存款中，“储户”通过“存取款单”和“储蓄所”发生联系。假定储户包括：账号，姓名，电话，地址，存款额；“储蓄所”包括：储蓄所编号，名称，电话，地址（假定一个储户可以在不同得储蓄所存取款）

**1、写出设计以上表格的语句（4分）**

**2、创建一个触发器TR1完成下面内容：**

**当向“存取款单”表中插入数据时，如果存取标志＝１则应该更改储户表让存款额加上存取金额，如果存取标志＝0则应该更改储户表让存款额减去存取金额,如果余额不足显示余额不足错误。**

CREATE TABLE CREATE TRIGGER tr1 on qukuan after insert

AS

BEGIN

declare @sid nvarchar(50)

declare @type int

declare @qian int

declare @yuer int

select @sid=sid,@type=[type],@m=m from inserted

select @yuer=yuer from cunkuan

if(@type=1)

begin

update cunkuan set yuer=yuer+@qian

end

else

begin

if(@yuer<@qian)

begin

print '余额不足'

end

else

begin

update cunkuan set yuer=yuer-@qian

end

end

END

GO

* 1. 本题用到下面三个关系表：

**CARD 借书卡: (CNO 卡号，NAME 姓名，CLASS 班级)**

**BOOKS 图书: (BNO 书号，BNAME 书名,AUTHOR 作者，PRICE 单价，QUANTITY 库存册数 )**

**BORROW 借书记录: (CNO 借书卡号，BNO 书号，RDATE 还书日期**

**备注：限定每人每种书只能借一本；库存册数随借书、还书而改变。**

**要求实现如下处理：**

**写出自定义函数，要求输入借书卡号能得到该卡号所借书金额的总和：**

CREATE FUNCTION getSUM

(

@CNO int

)

RETURNS int

AS

BEGIN

declare @sum int

select @sum=sum(price) from BOOKS where bno in (select bno from BORROW where cno=@CNO)

return @sum

END

GO

**找出借书超过5本的读者,输出借书卡号及所借图书册数。**

select CNO,count(BNO) as 借书数量from BORROW group by CNO having count(BNO)>3

**查询借阅了"水浒"一书的读者，输出姓名及班级。**

select name,class from card where cno in( select cno from borrow where bno in(

select bno from BOOKS where bname='水浒'))

**查询过期未还图书，输出借阅者（卡号）、书号及还书日期。**

select CNO,BNO,RDATE from borrow where getdate()>RDATE

**查询书名包括"网络"关键词的图书，输出书号、书名、作者。**

select bno,bname,author from books where bname like '网络%'

**查询现有图书中价格最高的图书，输出书名及作者。**

select bname,author from books where price in(select max(price) from books )

**查询当前借了"计算方法"但没有借"计算方法习题集"的读者，输出其借书卡号，并按卡号降序排序输出。**

select cno from borrow where bno in (select bno from books where bname='计算方法') and cno not in ( select cno from borrow where bno in(select bno from books where bname='计算方法习题集')) order by cno desc

或

SELECT a.CNO

FROM BORROW a,BOOKS b

WHERE a.BNO=b.BNO AND b.BNAME=N'计算方法'

AND NOT EXISTS(

SELECT \* FROM BORROW aa,BOOKS bb

WHERE aa.BNO=bb.BNO

AND bb.BNAME=N'计算方法习题集'

AND aa.CNO=a.CNO)

ORDER BY a.CNO DESC

**将"C01"班同学所借图书的还期都延长一周。**

update borrow set rdate=dateadd(day,7,rdate) from BORROW where cno in(select cno from card where class='一班')

**从BOOKS表中删除当前无人借阅的图书记录。**

DELETE A FROM BOOKS a

WHERE NOT EXISTS(

SELECT \* FROM BORROW

WHERE BNO=a.BNO)

**如果经常按书名查询图书信息，请建立合适的索引。**

CREATE CLUSTERED INDEX IDX\_BOOKS\_BNAME ON BOOKS(BNAME)

**在BORROW表上建立一个触发器，完成如下功能：如果读者借阅的书名是"数据库技术及应用"，就将该读者的借阅记录保存在BORROW\_SAVE表中（注ORROW\_SAVE表结构同BORROW表）**

CREATE TRIGGER TR\_SAVE ON BORROW

FOR INSERT,UPDATE

AS

IF @@ROWCOUNT>0

INSERT BORROW\_SAVE SELECT i.\*

FROM INSERTED i,BOOKS b

WHERE i.BNO=b.BNO

AND b.BNAME=N'数据库技术及应用'

**建立一个视图，显示"力01"班学生的借书信息（只要求显示姓名和书名）。**

CREATE VIEW V\_VIEW

AS

select name,bname

from books,card,borrow

where borrow.cno=card.cno and borrow.bno=books.bno and class='一班'

**查询当前同时借有"计算方法"和"组合数学"两本书的读者，输出其借书卡号，并按卡号升序排序输出。**

select a.cno from borrow a,borrow b

where a.cno=b.cno and

a.bno in(select bno from books where bname='计算方法') and

b.bno in(select bno from books where bname='组合数学')

order by a.cno desc

或

SELECT a.CNO

FROM BORROW a,BOOKS b

WHERE a.BNO=b.BNO

AND b.BNAME IN('计算方法','组合数学')

GROUP BY a.CNO

HAVING COUNT(\*)=2

ORDER BY a.CNO DESC

**用事务实现如下功能：一个借书卡号借走某书号的书，则该书的库存量减少1，当某书的库存量不够1本的时候，该卡号不能借该书**

alter PROCEDURE pro\_jieshu

@cno int,

@bno int,

@date datetime

AS

BEGIN

begin tran

declare @quantity int

select @quantity=quantity from books where bno=@bno

insert into borrow values(@cno,@bno,@date)

update books set quantity=@quantity-1 where bno=@bno

if(@quantity>0)

begin

commit tran

end

else

begin

print '已无库存'

rollback

end

END

GO

**用游标实现将书号为‘A001’的书本的价格提高10元**

declare @bno int

declare @bname nvarchar(50)

declare @author nvarchar(50)

declare @price int

declare @quantity int

declare mycursor cursor for select \* from books

open mycursor

fetch next from mycursor into @bno,@bname,@author,@price,@quantity

while(@@fetch\_status=0)

begin

if(@bno=2)

begin

update books set price=@price+10 where current of mycursor

end

fetch next from mycursor into @bno,@bname,@author,@price,@quantity

end

close mycursor

deallocate mycursor

**Student(S#,Sname,Sage,Ssex) 学生表**

**Course(C#,Cname,T#) 课程表**

**SC(S#,C#,score) 成绩表**

**Teacher(T#,Tname) 教师表**

**查询“001”课程比“002”课程成绩高的所有学生的学号；**

select a.S# from (select s#,score from SC where C#='001') a,(select s#,score

from SC where C#='002') b

where a.score>b.score and a.s#=b.s#;

**查询平均成绩大于60分的同学的学号和平均成绩；**

select S#,avg(score)

from sc

group by S# having avg(score) >**60**;

**查询所有同学的学号、姓名、选课数、总成绩；**

select Student.S#,Student.Sname,count(SC.C#),sum(score)

from Student left Outer join SC on Student.S#=SC.S#

group by Student.S#,Sname

**查询姓“李”的老师的个数；**

select count(distinct(Tname))

from Teacher

where Tname like '李%';

**查询没学过“叶平”老师课的同学的学号、姓名；**

select Student.S#,Student.Sname

from Student

where S# not in (select distinct( SC.S#) from SC,Course,Teacher where SC.C#=Course.C# and Teacher.T#=Course.T# and Teacher.Tname='叶平');

**查询学过“001”并且也学过编号“002”课程的同学的学号、姓名；**

select Student.S#,Student.Sname from Student,SC where Student.S#=SC.S# and SC.C#='001'and exists( Select \* from SC as SC\_2 where SC\_2.S#=SC.S# and SC\_2.C#='002');

**查询学过“叶平”老师所教的所有课的同学的学号、姓名；**

select S#,Sname

from Student

where S# in (select S# from SC ,Course ,Teacher where SC.C#=Course.C# and Teacher.T#=Course.T# and Teacher.Tname='叶平' group by S# having count(SC.C#)=(select count(C#) from Course,Teacher where Teacher.T#=Course.T# and Tname='叶平'));

**查询课程编号“002”的成绩比课程编号“001”课程低的所有同学的学号、姓名；**

Select S#,Sname from (select Student.S#,Student.Sname,score ,(select score from SC SC\_2 where SC\_2.S#=Student.S# and SC\_2.C#='002') score2

from Student,SC where Student.S#=SC.S# and C#='001') S\_2 where score2 <score;

**查询所有课程成绩小于60分的同学的学号、姓名；**

select S#,Sname

from Student

where S# not in (select Student.S# from Student,SC where S.S#=SC.S# and score>**60**);

**查询没有学全所有课的同学的学号、姓名；**

select Student.S#,Student.Sname

from Student,SC

where Student.S#=SC.S# group by Student.S#,Student.Sname having count(C#) <(select count(C#) from Course);

**查询至少有一门课与学号为“1001”的同学所学相同的同学的学号和姓名；**

select S#,Sname from Student,SC where Student.S#=SC.S# and C# in select C# from SC where S#='1001';

**查询至少学过学号为“001”同学所有一门课的其他同学学号和姓名；**

select distinct SC.S#,Sname

from Student,SC

where Student.S#=SC.S# and C# in (select C# from SC where S#='001');

**把“SC”表中“叶平”老师教的课的成绩都更改为此课程的平均成绩；**

update SC set score=(select avg(SC\_2.score)

from SC SC\_2

where SC\_2.C#=SC.C# ) from Course,Teacher where Course.C#=SC.C# and Course.T#=Teacher.T# and Teacher.Tname='叶平');

**查询和“1002”号的同学学习的课程完全相同的其他同学学号和姓名；**

select S# from SC where C# in (select C# from SC where S#='1002')

group by S# having count(\*)=(select count(\*) from SC where S#='1002');

**删除学习“叶平”老师课的SC表记录；**

Delete SC

from course ,Teacher

where Course.C#=SC.C# and Course.T#= Teacher.T# and Tname='叶平';

**向SC表中插入一些记录，这些记录要求符合以下条件：没有上过编号“003”课程的同学学号、2、号课的平均成绩；**

Insert SC select S#,'002',(Select avg(score)

from SC where C#='002') from Student where S# not in (Select S# from SC where C#='002');

**按平均成绩从高到低显示所有学生的“数据库”、“企业管理”、“英语”三门的课程成绩，按如下形式显示： 学生ID,,数据库,企业管理,英语,有效课程数,有效平均分**

SELECT S# as 学生ID

,(SELECT score FROM SC WHERE SC.S#=t.S# AND C#='004') AS 数据库

,(SELECT score FROM SC WHERE SC.S#=t.S# AND C#='001') AS 企业管理

,(SELECT score FROM SC WHERE SC.S#=t.S# AND C#='006') AS 英语

,COUNT(\*) AS 有效课程数, AVG(t.score) AS 平均成绩

FROM SC AS t

GROUP BY S#

ORDER BY avg(t.score)

**查询各科成绩最高和最低的分：以如下形式显示：课程ID，最高分，最低分**

SELECT L.C# As 课程ID,L.score AS 最高分,R.score AS 最低分

FROM SC L ,SC AS R

WHERE L.C# = R.C# and

L.score = (SELECT MAX(IL.score)

FROM SC AS IL,Student AS IM

WHERE L.C# = IL.C# and IM.S#=IL.S#

GROUP BY IL.C#)

AND

R.Score = (SELECT MIN(IR.score)

FROM SC AS IR

WHERE R.C# = IR.C#

GROUP BY IR.C#

);

**按各科平均成绩从低到高和及格率的百分数从高到低顺序**

SELECT t.C# AS 课程号,max(course.Cname)AS 课程名,isnull(AVG(score),**0**) AS 平均成绩

,**100** \* SUM(CASE WHEN isnull(score,**0**)>=**60** THEN **1** ELSE **0** END)/COUNT(\*) AS 及格百分数

FROM SC T,Course

where t.C#=course.C#

GROUP BY t.C#

ORDER BY **100** \* SUM(CASE WHEN isnull(score,**0**)>=**60** THEN **1** ELSE **0** END)/COUNT(\*) DESC

**查询如下课程平均成绩和及格率的百分数(用"1行"显示): 企业管理（001），马克（002），OO&UML （003），数据库（004）**

SELECT SUM(CASE WHEN C# ='001' THEN score ELSE **0** END)/SUM(CASE C# WHEN '001' THEN **1** ELSE **0** END) AS 企业管理平均分

,**100** \* SUM(CASE WHEN C# = '001' AND score >= **60** THEN **1** ELSE **0** END)/SUM(CASE WHEN C# = '001' THEN **1** ELSE **0** END) AS 企业管理及格百分数

,SUM(CASE WHEN C# = '002' THEN score ELSE **0** END)/SUM(CASE C# WHEN '002' THEN **1** ELSE **0** END) AS 马克思平均分

,**100** \* SUM(CASE WHEN C# = '002' AND score >= **60** THEN **1** ELSE **0** END)/SUM(CASE WHEN C# = '002' THEN **1** ELSE **0** END) AS 马克思及格百分数

,SUM(CASE WHEN C# = '003' THEN score ELSE **0** END)/SUM(CASE C# WHEN '003' THEN **1** ELSE **0** END) AS UML平均分

,**100** \* SUM(CASE WHEN C# = '003' AND score >= **60** THEN **1** ELSE **0** END)/SUM(CASE WHEN C# = '003' THEN **1** ELSE **0** END) AS UML及格百分数

,SUM(CASE WHEN C# = '004' THEN score ELSE **0** END)/SUM(CASE C# WHEN '004' THEN **1** ELSE **0** END) AS 数据库平均分

,**100** \* SUM(CASE WHEN C# = '004' AND score >= **60** THEN **1** ELSE **0** END)/SUM(CASE WHEN C# = '004' THEN **1** ELSE **0** END) AS 数据库及格百分数

FROM SC

**查询不同老师所教不同课程平均分从高到低显示**

SELECT max(Z.T#) AS 教师ID,MAX(Z.Tname) AS 教师姓名,C.C# AS 课程ＩＤ,MAX(C.Cname) AS 课程名称,AVG(Score) AS 平均成绩

FROM SC AS T,Course AS C ,Teacher AS Z

where T.C#=C.C# and C.T#=Z.T#

GROUP BY C.C#

ORDER BY AVG(Score) DESC

**查询如下课程成绩第 3 名到第 6 名的学生成绩单：企业管理（001），马克思（002），UML （003），数据库（004） [学生ID],[学生姓名],企业管理,马克思,UML,数据库,平均成绩**

SELECT DISTINCT top **3**

SC.S# As 学生学号,

Student.Sname AS 学生姓名 ,

T1.score AS 企业管理,

T2.score AS 马克思,

T3.score AS UML,

T4.score AS 数据库,

ISNULL(T1.score,**0**) + ISNULL(T2.score,**0**) + ISNULL(T3.score,**0**) + ISNULL(T4.score,**0**) as 总分

FROM Student,SC LEFT JOIN SC AS T1

ON SC.S# = T1.S# AND T1.C# = '001'

LEFT JOIN SC AS T2

ON SC.S# = T2.S# AND T2.C# = '002'

LEFT JOIN SC AS T3

ON SC.S# = T3.S# AND T3.C# = '003'

LEFT JOIN SC AS T4

ON SC.S# = T4.S# AND T4.C# = '004'

WHERE student.S#=SC.S# and

ISNULL(T1.score,**0**) + ISNULL(T2.score,**0**) + ISNULL(T3.score,**0**) + ISNULL(T4.score,**0**)

NOT IN

(SELECT

DISTINCT

TOP **15** WITH TIES

ISNULL(T1.score,**0**) + ISNULL(T2.score,**0**) + ISNULL(T3.score,**0**) + ISNULL(T4.score,**0**)

FROM sc

LEFT JOIN sc AS T1

ON sc.S# = T1.S# AND T1.C# = 'k1'

LEFT JOIN sc AS T2

ON sc.S# = T2.S# AND T2.C# = 'k2'

LEFT JOIN sc AS T3

ON sc.S# = T3.S# AND T3.C# = 'k3'

LEFT JOIN sc AS T4

ON sc.S# = T4.S# AND T4.C# = 'k4'

ORDER BY ISNULL(T1.score,**0**) + ISNULL(T2.score,**0**) + ISNULL(T3.score,**0**) + ISNULL(T4.score,**0**) DESC);

**统计列印各科成绩,各分数段人数:课程ID,课程名称,[100-85],[85-70],[70-60],[ <60]**

SELECT SC.C# as 课程ID, Cname as 课程名称

,SUM(CASE WHEN score BETWEEN **85** AND **100** THEN **1** ELSE **0** END) AS [100 - 85]

,SUM(CASE WHEN score BETWEEN **70** AND **85** THEN **1** ELSE **0** END) AS [85 - 70]

,SUM(CASE WHEN score BETWEEN **60** AND **70** THEN **1** ELSE **0** END) AS [70 - 60]

,SUM(CASE WHEN score < **60** THEN **1** ELSE **0** END) AS [60 -]

FROM SC,Course

where SC.C#=Course.C#

GROUP BY SC.C#,Cname;

**查询学生平均成绩及其名次**

SELECT **1**+(SELECT COUNT( distinct 平均成绩)

FROM (SELECT S#,AVG(score) AS 平均成绩

FROM SC

GROUP BY S#

) AS T1

WHERE 平均成绩 > T2.平均成绩) as 名次,

S# as 学生学号,平均成绩

FROM (SELECT S#,AVG(score) 平均成绩

FROM SC

GROUP BY S#

) AS T2

ORDER BY 平均成绩 desc;

**查询各科成绩前三名的记录:(不考虑成绩并列情况)**

SELECT t1.S# as 学生ID,t1.C# as 课程ID,Score as 分数

FROM SC t1

WHERE score IN (SELECT TOP **3** score

FROM SC

WHERE t1.C#= C#

ORDER BY score DESC

)

ORDER BY t1.C#;

**查询每门课程被选修的学生数**

select c#,count(S#) from sc group by C#;

**查询出只选修了一门课程的全部学生的学号和姓名**

select SC.S#,Student.Sname,count(C#) AS 选课数

from SC ,Student

where SC.S#=Student.S# group by SC.S# ,Student.Sname having count(C#)=**1**;

**查询男生、女生人数**

Select count(Ssex) as 男生人数 from Student group by Ssex having Ssex='男';

Select count(Ssex) as 女生人数 from Student group by Ssex having Ssex='女'；

**查询姓“张”的学生名单**

SELECT Sname FROM Student WHERE Sname like '张%';

**查询同名同性学生名单，并统计同名人数**

select Sname,count(\*) from Student group by Sname having count(\*)>**1**;

**1981年出生的学生名单(注：Student表中Sage列的类型是datetime)**

select Sname, CONVERT(char (**11**),DATEPART(year,Sage)) as age

from student

where CONVERT(char(**11**),DATEPART(year,Sage))='1981';

**查询每门课程的平均成绩，结果按平均成绩升序排列，平均成绩相同时，按课程号降序排列**

Select C#,Avg(score) from SC group by C# order by Avg(score),C# DESC ;

**查询平均成绩大于85的所有学生的学号、姓名和平均成绩**

select Sname,SC.S# ,avg(score)

from Student,SC

where Student.S#=SC.S# group by SC.S#,Sname having avg(score)>**85**;

**查询课程名称为“数据库”，且分数低于60的学生姓名和分数**

Select Sname,isnull(score,**0**)

from Student,SC,Course

where SC.S#=Student.S# and SC.C#=Course.C# and Course.Cname='数据库'and score <**60**;

**查询所有学生的选课情况；**

SELECT SC.S#,SC.C#,Sname,Cname

FROM SC,Student,Course

where SC.S#=Student.S# and SC.C#=Course.C# ;

**查询任何一门课程成绩在70分以上的姓名、课程名称和分数；**

SELECT distinct student.S#,student.Sname,SC.C#,SC.score

FROM student,Sc

WHERE SC.score>=**70** AND SC.S#=student.S#;

**查询不及格的课程，并按课程号从大到小排列**

select c# from sc where scor e <**60** order by C# ;

**查询课程编号为003且课程成绩在80分以上的学生的学号和姓名；**

select SC.S#,Student.Sname from SC,Student where SC.S#=Student.S# and Score>**80** and C#='003';

**求选了课程的学生人数**

select count(\*) from sc;

**查询选修“叶平”老师所授课程的学生中，成绩最高的学生姓名及其成绩**

select Student.Sname,score from Student,SC,Course C,Teacher

where Student.S#=SC.S# and SC.C#=C.C# and C.T#=Teacher.T# and Teacher.Tname='叶平' and SC.score=(select max(score)from SC where C#=C.C# );

**查询各个课程及相应的选修人数**

select count(\*) from sc group by C#;

**查询不同课程成绩相同的学生的学号、课程号、学生成绩**

select distinct A.S#,B.score from SC A ,SC B where A.Score=B.Score and A.C# <>B.C# ;

**查询每门功成绩最好的前两名**

SELECT t1.S# as 学生ID,t1.C# as 课程ID,Score as 分数

FROM SC t1

WHERE score IN (SELECT TOP **2** score

FROM SC

WHERE t1.C#= C#

ORDER BY score DESC

)

ORDER BY t1.C#;

**统计每门课程的学生选修人数（超过10人的课程才统计）。要求输出课程号和选修人数，查询结果按人数降序排列，查询结果按人数降序排列，若人数相同，按课程号升序排列**

select C# as 课程号,count(\*) as 人数

from sc

group by C#

order by count(\*) desc,c#

**检索至少选修两门课程的学生学号**

select S#

from sc

group by s#

having count(\*) > = **2**

**查询全部学生都选修的课程的课程号和课程名**

select C#,Cname

from Course

where C# in (select c# from sc group by c#)

**查询没学过“叶平”老师讲授的任一门课程的学生姓名**

select Sname from Student where S# not in (select S# from Course,Teacher,SC where Course.T#=Teacher.T# and SC.C#=course.C# and Tname='叶平');

**查询两门以上不及格课程的同学的学号及其平均成绩**

select S#,avg(isnull(score,**0**)) from SC where S# in (select S# from SC where score <**60** group by S# having count(\*)>**2**)group by S#;

**检索“004”课程分数小于60，按分数降序排列的同学学号**

select S# from SC where C#='004'and score <**60** order by score desc;

**删除“002”同学的“001”课程的成绩**

delete from Sc where S#='001'and C#='001';

1. 第4阶段：ASP.NET高级
   1. Application 、Cookie和 Session 三种会话有什么不同？

Application是用来存取整个网站全局的信息，而Session是用来存取与具体某个访问者关联的信息。Cookie是保存在客户端的，机密信息不能保存在Cookie中，只能放小数据；Session是保存在服务器端的，比较安全，可以放大数据。

* 1. Session有什么重大BUG，微软提出了什么方法加以解决？

IIS中由于有进程回收机制，系统繁忙的话Session会丢失，IIS重启也会造成Session失。这样用户就要重新登录或者重新添加购物车、验证码等放到Session中的信息。可以用State Server或SQL Server数据库的方式存储Session不过这种方式比较慢，而且无法捕获Session的END事件。但是这不是Bug，只能说是In-Proc方式存储Session的缺陷，缺陷是和Bug不一样的，In-Proc方式存储Session会由服务器来决定什么时候释Session，In-Proc方式不满足要求的话完全可以用StateServer和数据库的方式。

StateServer还可以解决集群Session共享的问题。

* 1. Asp.net中的错误机制？

1）定制错误页来将显示一个友好的报错页面。

2）页面中未捕获一样会触发Page\_Error，应用程序中的未捕获异常会触发Application\_Error。通过HttpContext.Current.Server.GetLastError()拿到未捕捉异常，记录到Log4Net日志中。

* 1. 介绍几个使用过的开源的项目？

Lucene.net、NPOI、JQuery、ASP.Net AJAX toolkit、Quartz.Net、JqueryUI、ASP.Net MVC。在CodePlex、SourceForge等网站上有更多的开源项目。

* 1. 如何提高页面的显示速度？假如一个页面的加载时间是10.89s，你会用什么样的方式进行优化？

首先要找出问题所在，是服务器端运行的速度慢还是服务器端到客户端的下载慢还是页面在浏览器的加载速度慢。

如果是服务器端运行速度慢，则找是数据库的原因还是算法的问题，如果是数据库的问题则尝试添加索引、优化SQL语句，如果是算法的问题，则优化算法。如果对于一些不经常改动的页面可以使用静态页技术！对于一些数据不需要及时更新的而且取数据的过程非常耗时可以使用缓存。页面中的内容可以按需加载（比如说可以像verycd网站的评论那样当用户需要看的时候再加载其内容），可以在图片需要显示的时候再进行加载。

如果是服务器端到客户端的下载慢则看是页面体积过于臃肿还是网络问题，如果是页面体积过于臃肿，则优化HTML代码，去掉无用的标签，压缩JS、CSS，可以用CSS Spirit技术将多个图片放到一个图片中，减少向服务器的请求。如果是网络问题，则尝试在不同的网络、地区部署服务器，然后使用CDN技术加速访问。

如果是页面中的JavaScript运行复杂导致运行速度慢，则优化JavaScript。

webkaka：测网速。

* 1. 在程序编码的时候，你会对SQL注入漏洞的防范采取什么样的措施？

尽量不要拼SQL语句！使用参数化查询或存储过程可以防止SQL注入攻击！在必须用SQL拼接的地方对用户输入的内容进行检查、过滤。

* 1. 你会采用什么样的策略和方法来实现系统缓存？

在ASP.NET中 缓存有 页面缓存，数据源缓存，和一些自己定义的缓存！

对于那些整个页面不经常变化的我们可以使用页面缓存，而对于那些执行非常耗时的SQL语句并且数据的及时性要求不高的我们可以使用数据源缓存。

对于页面缓存、数据源缓存等都不满足要求的情况下采用直接操作HttpRuntime.Cache来自定义缓存策略。如果需要用多台Web服务器作为一个集群来承载网站的情况，则可以部署专门的分布式缓存服务器，比如Memcached。

* 1. 网站想要实现文件防盗链的功能，说说你的解决方案？

读取HTTP报文头中的UrlReferrer在Application\_BeginRequest中我们可以判断用户的请求是否来源于本网站。如果不是我们可以终止用户的请求（Response.End();）！

* 1. 进程和线程的区别？

进程是系统进行资源分配和调度的单位；线程是CPU调度和分派的单位，一个进程可以有多个线程，这些线程共享这个进程的资源。

* 1. Ajax操作怎么处理session过期？

当Session过期时，我们获取请求头信息值判断是否为ajax请求，如果是，我们可以返回特定格式的JSON数据，客户端可以对此数据处理，发现session失效，可以跳转到其他页面如登录等。

* 1. 简要谈一下你对微软.Net架构下remoting和webservice两项技术的理解及实际中的应用？

Remoting可以利用TCP/IP，二进制传送提高效率，webservice可利用http，穿透防火墙。

远程逻辑调用，remoing接口只能用在.net remoting是.net 中用来跨越machine,process, appdomain 进行方法调用的技术,对于三成结构的程序，就可以使用remoting技术来构建，它是分布应用的基础技术.相当于以前的DCOM

WebService是一种构建应用程序的普通模型，并能在所有支持internet网通讯的操作系统上实施。Web Service令基于组件的开发和web的结合达到最佳，基于组件的对象模型。

* 1. 序列化有何作用？

通过流类型可以方便地操作各种字节流，但如何把现有的实例对象转换为方便传输的字节流，就需要用到序列化的技术。

* 1. .Net中会存在内存泄漏吗？请简单描述。

所谓内存泄露就是指一个不再被程序使用的对象或变量一直被占据在内存中。.Net中有垃圾回收机制，它可以保证一对象不再被引用的时候，即对象编程了孤儿的时候，对象将自动被垃圾回收器从内存中清除掉。虽然.Net可以回收无用的对象，但是.Net仍然存在由于使用不当导致的内存泄露问题。.Net中的内存泄露的情况：长生命周期的对象持有短生命周期对象的引用就很可能发生内存泄露，尽管短生命周期对象已经不再需要，但是因为长生命周期对象持有它的引用而导致不能被回收，这就是.Net中内存泄露的发生场景，通俗地说，就是程序员可能创建了一个对象，以后一直不再使用这个对象，这个对象却一直被引用，即这个对象无用但是却无法被垃圾回收器回收的，这就是.Net中可能出现内存泄露的情况，例如，缓存系统，我们加载了一个对象放在缓存中(例如放在一个全局Dictionary对象中)，然后一直不再使用它，这个对象一直被缓存引用，但却不再被使用。

* 1. 请编程遍历WinForm页面上所有TextBox控件并给它赋值为string.Empty？

foreach (System.Windows.Forms.Control control in this.Controls)

{

if (control is System.Windows.Forms.TextBox)

{

System.Windows.Forms.TextBox tb = (System.Windows.Forms.TextBox)control ;

tb.Text = String.Empty ;

}

}

* 1. 触发器的作用？

触发器是一中特殊的存储过程，主要是通过事件来触发而被执行的。它可以强化约束，来维护数据的完整性和一致性，可以跟踪数据库内的操作从而不允许未经许可的更新和变化。可以联级运算。如，某表上的触发器上包含对另一个表的数据操作，而该操作又会导致该表触发器被触发。

* 1. 索引的作用？和它的优点缺点是什么？

索引就一种特殊的查询表，数据库的搜索引擎可以利用它加速对数据的检索。它很类似与现实生活中书的目录，不需要查询整本书内容就可以找到想要的数据。索引可以是唯一的，创建索引允许指定单个列或者是多个列。缺点是它减慢了数据录入的速度，同时也增加了数据库的尺寸大小。

* 1. 说一下SQLServer中索引的两种类型？

聚簇(或者叫做聚集，cluster)索引和非聚簇索引。

字典的拼音目录就是聚簇(cluster)索引，笔画目录就是非聚簇索引。这样查询“G到M的汉字”就非常快，而查询“6划到8划的字”则慢。

聚簇索引是一种特殊索引，它使数据按照索引的排序顺序存放表中。聚簇索引类似于字典，即所有词条在字典中都以字母顺序排列。聚簇索引实际上重组了表中的数据，所以你只能在表中建立一个聚簇索引。

当数据按值的范围查询时，聚簇索引就显得特别有用。因为所有SQLServer都必需先找到所查询范围的第一行，然后依次下去，直到该范围的最后一个值找到为止，并且保证了所有其他值也落在这个范围内。举一个例子，一个应用程序要查找首字母位于G和P之间的姓名列表，SQLServer首先找到以字母G开头的名字，取出所有记录，直到找到以字母P开头的名字为止，这种方法使得查询过程非常高效。

进行大量数据改动的表不适宜用聚簇索引，因为SQLServer将不得不在表中维护行的次序。如果要索引的值极少，例如一个列包含的全都是1和0，创建聚簇索引就不是个好主意。如果表经常由一个指定的列来排序，该列将是簇索引的最佳候选列。这是因为表中的数据已经为你排好序了。如果访问一个表并使用BETWEEN、<、>、>=或<=操作符来返回一个范围的值时，应该考虑使用聚簇索引。

* 1. 什么是事务？什么是锁？

事务就是被绑定在一起作为一个逻辑工作单元的SQL语句分组，如果任何一个语句操作失败那么整个操作就被失败，以后操作就会回滚到操作前状态，或者是上有个节点。为了确保要么执行，要么不执行，就可以使用事务。要将有组语句作为事务考虑，就需要通过ACID测试，即原子性，一致性，隔离性和持久性。

锁：在所以的DBMS中，锁是实现事务的关键，锁可以保证事务的完整性和并发性。与现实生活中锁一样，它可以使某些数据的拥有者，在某段时间内不能使用某些数据或数据结构。当然锁还分级别的。

* 1. 什么叫视图？游标是什么？

视图是一种虚拟的表，具有和物理表相同的功能。可以对视图进行增，改，查，操作，试图通常是有一个表或者多个表的行或列的子集。对视图的修改不影响基本表。它使得我们获取数据更容易，相比多表查询。

游标：是对查询出来的结果集作为一个单元来有效的处理。游标可以定在该单元中的特定行，从结果集的当前行检索一行或多行。可以对结果集当前行做修改。一般不使用游标，但是需要逐条处理数据的时候，游标显得十分重要。

* 1. 什么是SQL注入式攻击？

所谓SQL注入式攻击，就是攻击者把SQL命令插入到Web表单的输入域或页面请求的查询字符串，欺骗服务器执行恶意的SQL命令。在某些表单中，用户输入的内容直接用来构造（或者影响）动态SQL命令，或作为存储过程的输入参数，这类表单特别容易受到SQL注入式攻击。常见的SQL注入式攻击过程类如：

⑴ 某个ASP.NET Web应用有一个登录页面，这个登录页面控制着用户是否有权访问应用，它要求用户输入一个名称和密码。

⑵ 登录页面中输入的内容将直接用来构造动态的SQL命令，或者直接用作存储过程的参数。下面是ASP.NET应用构造查询的一个例子：

System.Text.StringBuilder query = new System.Text.StringBuilder(

"SELECT \* from Users WHERE login = '")

.Append(txtLogin.Text).Append("' AND password='")

.Append(txtPassword.Text).Append("'");

⑶ 攻击者在用户名字和密码输入框中输入"'或'1'='1"之类的内容。

⑷ 用户输入的内容提交给服务器之后，服务器运行上面的ASP.NET代码构造出查询用户的SQL命令，但由于攻击者输入的内容非常特殊，所以最后得到的SQL命令变成：SELECT \* from Users WHERE login = '' or '1'='1' AND password = '' or '1'='1'。

⑸ 服务器执行查询或存储过程，将用户输入的身份信息和服务器中保存的身份信息进行对比。

⑹ 由于SQL命令实际上已被注入式攻击修改，已经不能真正验证用户身份，所以系统会错误地授权给攻击者。

如果攻击者知道应用会将表单中输入的内容直接用于验证身份的查询，他就会尝试输入某些特殊的SQL字符串篡改查询改变其原来的功能，欺骗系统授予访问权限。

系统环境不同，攻击者可能造成的损害也不同，这主要由应用访问数据库的安全权限决定。如果用户的帐户具有管理员或其他比较高级的权限，攻击者就可能对数据库的表执行各种他想要做的操作，包括添加、删除或更新数据，甚至可能直接删除表。

* 1. 如何防范SQL注入式攻击？

好在要防止ASP.NET应用被SQL注入式攻击闯入并不是一件特别困难的事情，只要在利用表单输入的内容构造SQL命令之前，把所有输入内容过滤一番就可以了。过滤输入内容可以按多种方式进行。

⑴ 对于动态构造SQL查询的场合，可以使用下面的技术：

第一：替换单引号，即把所有单独出现的单引号改成两个单引号，防止攻击者修改SQL命令的含义。再来看前面的例子，"SELECT \* from Users WHERE login = ''' or ''1''=''1' AND password = ''' or ''1''=''1'"显然会得到与"SELECT \* from Users WHERE login = '' or '1'='1' AND password = '' or '1'='1'"不同的结果。

第二：删除用户输入内容中的所有连字符，防止攻击者构造出类如"SELECT \* from Users WHERE login = 'mas' -- AND password =''"之类的查询，因为这类查询的后半部分已经被注释掉，不再有效，攻击者只要知道一个合法的用户登录名称，根本不需要知道用户的密码就可以顺利获得访问权限。

第三：对于用来执行查询的数据库帐户，限制其权限。用不同的用户帐户执行查询、插入、更新、删除操作。由于隔离了不同帐户可执行的操作，因而也就防止了原本用于执行SELECT命令的地方却被用于执行INSERT、UPDATE或DELETE命令。

⑵ 用存储过程来执行所有的查询。SQL参数的传递方式将防止攻击者利用单引号和连字符实施攻击。此外，它还使得数据库权限可以限制到只允许特定的存储过程执行，所有的用户输入必须遵从被调用的存储过程的安全上下文，这样就很难再发生注入式攻击了。

⑶ 限制表单或查询字符串输入的长度。如果用户的登录名字最多只有10个字符，那么不要认可表单中输入的10个以上的字符，这将大大增加攻击者在SQL命令中插入有害代码的难度。

⑷ 检查用户输入的合法性，确信输入的内容只包含合法的数据。数据检查应当在客户端和服务器端都执行——之所以要执行服务器端验证，是为了弥补客户端验证机制脆弱的安全性。

在客户端，攻击者完全有可能获得网页的源代码，修改验证合法性的脚本（或者直接删除脚本），然后将非法内容通过修改后的表单提交给服务器。因此，要保证验证操作确实已经执行，唯一的办法就是在服务器端也执行验证。你可以使用许多内建的验证对象，例如 RegularExpressionValidator，它们能够自动生成验证用的客户端脚本，当然你也可以插入服务器端的方法调用。如果找不到现成的验证对象，你可以通过CustomValidator自己创建一个。

⑸ 将用户登录名称、密码等数据加密保存。加密用户输入的数据，然后再将它与数据库中保存的数据比较，这相当于对用户输入的数据进行了"消毒"处理，用户输入的数据不再对数据库有任何特殊的意义，从而也就防止了攻击者注入SQL命令。 System.Web.Security.FormsAuthentication类有一个HashPasswordForStoringInConfigFile，非常适合于对输入数据进行消毒处理。

⑹ 检查提取数据的查询所返回的记录数量。如果程序只要求返回一个记录，但实际返回的记录却超过一行，那就当作出错处理。

* 1. 事务是什么？

事务是作为一个逻辑单元执行的一系列操作，一个逻辑工作单元必须有四个属性，称为 ACID（原子性、一致性、隔离性和持久性）属性，只有这样才能成为一个事务：

1)**原子性**

事务必须是原子工作单元；对于其数据修改，要么全都执行，要么全都不执行。

2)**一致性**

事务在完成时，必须使所有的数据都保持一致状态。在相关数据库中，所有规则都必须应用于事务的修改，以保持所有数据的完整性。事务结束时，所有的内部数据结构（如 B 树索引或双向链表）都必须是正确的。

3)**隔离性**

由并发事务所作的修改必须与任何其它并发事务所作的修改隔离。事务查看数据时数据所处的状态，要么是另一并发事务修改它之前的状态，要么是另一事务修改它之后的状态，事务不会查看中间状态的数据。这称为可串行性，因为它能够重新装载起始数据，并且重播一系列事务，以使数据结束时的状态与原始事务执行的状态相同。

4)**持久性**

事务完成之后，它对于系统的影响是永久性的。该修改即使出现系统故障也将一直保持。

* 1. Ajax原理？

简述： Ajax的原理就是:通过javascript的方式，将前台数据通过xmlhttp对象传递到后台，后台在接收到请求后，将需要的结果，再传回到前台，这样就可以实现不需要页面的回发，页是数据实现来回传递，从页实现无刷新。

Ajax的原理简单来说，实际上就是通过XmlHttpRequest对象来向服务器发异步请求，从服务器获得数据，然后用javascript来操作DOM而更新页面。

这其中最关键的一步就是从服务器获得请求数据。要清楚这个过程和原理，我们必须对 XMLHttpRequest有所了解。

总结：我们可以看出，XMLHttpRequest对象完全用来向服务器发出一个请求的，它的作用也局限于此，但它的作用是整个ajax实现的关键，我们可以把服务器端看成一个数据接口，它返回的是一个纯文本流，当然，这个文本流可以是XML格式，可以是Html，可以是Javascript代码，也可以只是一个字符串。这时候，XMLHttpRequest向服务器端请求这个页面，服务器端将文本的结果写入页面，这和普通的web开发流程是一样的，不同的是，客户端在异步获取这个结果后，不是直接显示在页面，而是先由javascript来处理，然后再显示在页面。

1. 第5阶段：ASP.NET企业级技术
   1. 谈谈你对MVC和三层架构的理解？

MVC即模型、视图、控制器，模型表示业务数据及业务处理，用来封装数据及行为；视图是用户看到并与之交互的界面；控制器接受用户输入并调用模型和视图去完成用户的请求。使用MVC有利于关注点分离，自动化UI测试成为了可能。

三层架构即表现层(UI)、业务逻辑层(BLL)、数据访问层(DAL)。区分层次的目的即为了“高内聚，低耦合”的思想。表现层通俗讲就是展现给用户的界面，业务逻辑层即针对具体问题的操作，也可以说是对数据层的操作，对数据业务逻辑处理。数据访问层：该层所做事务直接操作数据库，针对数据的增添、删除、修改、更新、查找等。

* 1. 在MVC中如何保持Sessions？

可使用tempdata、viewdata、viewbag三种方式。其中tempdata：在不同的控制器或动作间转换时保持数据。另外，进行页面转向时，tempdata可以保持数据，它是一个内部的Session变量。Viewdata：可以在控制器和视图间保持数据。Viewbag：它是视图数据的动态包装，使用viewbag不需要类型转换，它使用的是内部动态关键词。

* 1. MVC中如何做输入验证？

在M层使用数据模型的属性标签形如[Required],在C层使用ModelState.IsValid属性检查数据是否正确，也可在C层使用JavaScript进行数据验证。

* 1. 已经有了ASPX，为什么还要Razor？哪个更好？

Razor是一个轻量级的视图引擎，MVC3引入，相比ASPX，Razor更干净、轻量级、语法更简单，只需要使用@关键字，[如@DataTime.Now](mailto:如@DataTime.Now)

* 1. MVC有多少种不同类型的结果类型，请详细描述？

有12种，最主要的是ActionResult类，它是一个基础类，它有11个子类型。

**ViewResult** - 给响应流渲染指定的视图

**PartialViewResult** - 给响应流渲染指定的局部视图

**EmptyResult** - 返回空的响应结果。

**RedirectResult** - 执行一个HTTP转向到指定的URL。

**RedirectToRouteResult** -执行一个HTTP转向到一个URL，这个URL由基于路由数据的路由引擎来决定

**JsonResult** - 序列化一个ViewData对像到JSON格式。

**JavaScriptResult** - 返回一段javascript代码，它可以在客户端执行。

**ContentResult** - 写内容到响应流，不需要视图支持。

**FileContentResult** - 返回一个文件到客户端。

**FileStreamResult** - 返回一个文件到客户端，它提供的是流。

**FilePathResult** - 返回一个文件到客户端。

* 1. 列举你用过的javascript框架，并简述它们的优缺点？

**js框架**：jQuery EasyUI、ExtJS、Bootstrap、AngularJS等等。

**jQuery EasyUI**：轻量级web前端ui开发框架，尤其适合MIS系统的界面开发，能够重用jquery插件。

**ExtJS**：统一的前端UI开发框架，学习难度中等。尤其适合MIS系统的界面开发，开发文档和例子代码都比较完整。缺点是大量的js脚本，降低了运行速度。

**Bootstrap**：响应式网站开发框架，优点是降低了后端开发人员开发前端页面的难度，统一了界面风格，缺点是界面风格比较单一。

**AngularJS**：将java后端的优秀特性引入到了js前端，大而全的框架。缺点是学习曲线高，Angular2几乎重写。

* 1. 什么是反射？

程序集包含模块，而模块又包括类型，类型下有成员，反射就是管理程序集，模块，类型的对象，它能够动态的创建类型的实例，设置现有对象的类型或者获取现有对象的类型，能调用类型的方法和访问类型的字段属性。它是在运行时创建和使用类型实例。

* 1. XML 与 HTML 的主要区别？

1. XML是区分大小写字母的，HTML不区分。

2. 在HTML中，如果上下文清楚地显示出段落或者列表键在何处结尾，那么你可以省略</p>或者</li>之类的结束 标记。在XML中，绝对不能省略掉结束标记。

HTML：<img src="1.jpg"><br><br>

XML：<img src="1.jpg"></img><br/><br/>

3. 在XML中，拥有单个标记而没有匹配的结束标记的元素必须用一个 / 字符作为结尾。这样分析器就知道不用 查找结束标记了。

4. 在XML中，属性值必须分装在引号中。在HTML中，引号是可用可不用的。

5. 在HTML中，可以拥有不带值的属性名。在XML中，所有的属性都必须带有相应的值。

XML是用来存储和传输数据的

HTML是用来显示数据的

如果使用了完全符合XML语法要求的HTML，那么就叫做符合XHTML标准。符合XHTML标准的页面有利于SEO。

* 1. MVC（模型、视图、控制器）架构的控制流程

所有的终端用户请求被发送到控制器。

控制器依赖请求去选择加载哪个模型，并把模型附加到对应的视图。

附加了模型数据的最终视图做为响应发送给终端用户。

* 1. MVC同时适用于Windows应用和Web应用吗?

相比Windows应用，MVC架构更适用于Web应用。对于Windows应用，MVP(Model View Presenter)架构更好一点 。如果你使用WPF和Silverlight，MVVM更适合。

* 1. 使用MVC有哪些好处?

MVC有两个大的好处：

离了关注点。后台代码被移到单独的类文件，我们可以最大限度的重复利用代码。

自动化UI测试成为可能，因为后台代码移到了.NET类。这让我们更容易做单元测试和自动化测试

* 1. “HTML.TextBox” 和 “HTML.TextBoxFor”有什么不同?

它们两个输出相同的HTML， “HTML.TextBoxFor”是强类型的，但“HTML.TextBox”不是。

* 1. MVC的路由选择是什么?

路由选择功能帮你定义一个URL规则，映射URL到控制器。

* 1. 在哪里写路由映射表？

在 “global.asax” 文件。

* 1. 我们可以映射多个URL到同一个动作吗？

是的，可以。只需要添加多条不同Key名字的记录，并且指定同样的控制器和动作。

* 1. 使用hyperlink生成链接，如何从一个视图链接到另一个视图?

使用ActionLink方法，如下图所示。下面的代码生成一个简单的URL，链接到"Home"控制器的GotoHome动作。<%= Html.ActionLink("Home","Gotohome") %>

* 1. 如何限制一个动作的类型为GET或POST?

我们可以给MVC的动作一个HttpGet或HttpPost属性去限制HTTP的类型。

* 1. 在MVC中如何保持Sessions?

可以通过三种方式保持： tempdata, viewdata, 和viewbag。

* 1. tempdata, viewdata, 和 viewbag之间有什么不同?

Tempdata -在不同的控制器或动作间转换时保持数据。另外，进行页面转向时，tempdata可以保持数据。它 是一个内部的Session变量。

Viewdata - 可以在控制器和视图间保持数据。

ViewBag - 它是视图数据的动态包装。使用Viewbag不需要类型转换。它使用的是内部动态关健词。

* 1. MVC的局部视图是什么?

局部视图是一个可重复调用的视图（和用户控件一样），它可以嵌入到视图里面。

* 1. ASP.NET MVC是如何运行的？

当一个用户请求提交上来后，ASP.NET MVC会针对当前请求实施路由解析，解析的目标就是找到用户需要的 Controller并激活它，执行对应的Action方法，最终返回用户需要的东西。

　　也就是说，当ASP.NET MVC接收到抵达的请求后，首要任务就是通过该请求解析得到对应的Controller和 Action名称，那么它是如何解析的呢？这就要了解一下Asp.Net MVC的路由系统了。

* 1. 什么是ASP.NET MVC？

一句话概括：ASP.NET MVC就是建立在ASP.NET平台上，基于MVC模式的，Web应用框架。

详细点说：ASP.NET平台采用管道式设计，具有良好的扩展性。整个ASP.NET MVC框架就是通过自定义 ASP.NET的HttpModule和HttpHandler这两个核心组件而建立的。

　　注：MVC中的Model主要体现为维持应用状态并提供业务功能，但ASP.NET MVC中的Model与之是不同的，后者 仅仅是绑定到View上的数据而已，两者并不是一回事，在学习理解过程中要格外注意，不要混淆。