گزارش کارآموزی در یک شرکت برنامه نویسی

ارسال شده توسط مرتضی شبان برای انتشار در پروژه دات کام
www.Prozhe.ocm

فصل اول

۱-معرفی شرکت

زمینه های فعالیت شرکت نرم افزار گستر نوآئین:

تولید و عرضه انواع نرم افزارهای رایانه ای تولید نرم افزارهای تجاری،حسابداری، انبارداری و...

تولید نرم افزارهای مولتی مدیا(چند رسانه ای)

طراحی وب سایت های ایستا و پویا، فلش و ...

شرکت نرم افزار گستر نوآئین با بکارگیری و استفاده از نیروهای متخصص و با تجربه در امر برنامه نویسی رایانه و تولید صفحات وب درصدد است تا با ارائه محصولاتی نوین، کارآمد و ایمن به کاربران گرامی، گامی موثر در جهت برآورده شدن نیازهای نرم افزاری جامعه و رشد و توسعه ی تکنولوژی در کشور بردارد. امید است با سعی و تلاش خود بتوانیم در توسعه فرهنگ استفاده از رایانه در کشور و برآوردن نیازهای کاربران و استفاده کنندگان از تولیدات نرم افزاری سهیم باشیم.

زمینه های فعالیت

برنامه نویسی

طراحی سایت

مولتی مدیا

پشتیبانی

سامانه پیام کوتاه

۱-۱ درآمد تقریبی

سالانه ۲۰ میلیون تومان می باشد

مثلا یکی از قراردادها که برای تولید یک سایت بسته شد، که در طول ۴ ماه باید تحویل داده شود به مبلغ ۴ میلیون تومان بسته شد.

فصل دوم

۲- کارهای انجام شده در دوره کارآموزی

2-1آشنایی با محیط C#

من در دوره کارآموزی با محیط C# آشنا شدم و کارهایی را در این محیط انجام داده ام که در زیر به برخی از آنها اشاره کرده ام

نوع های داده ای: اعداد صحیح و اعداد اعشاری و اعداد اعشاری با دقت معمولی

کار با رشته ها: اتصال رشته ها، استفاده از عملگر اتصال رشته در درون برنامه ، زیر رشته ها، قالب بندی رشته ها، جایگزینی زیر رشته ها، تبدیل نوع های داده ای

کار با تاریخها: قالب بندی تاریخها ، استفاده از خاصیتهای DateTime

دستورات شرطی و حلقه ها

۲-۲ طراحی فرم

فرمهای برنامه راطراحی کردم

طراحی در برنامه نویسی بسیار مهم است و باید با دقت انجام شود برای قرار دادن گزینه های مورد نظر در فرم می توان اشیاء را از قسمت tools انتخاب کرد و برای تنظیم آنها می توان از پنجره properties استفاده کرد

2-3رویدادها در C#

در ضمن با رویدادهای صفحه در محیط برنامه نویسی C# کار کردم

برای کار با رویدادها باید در صفحه ی فرم برنامه بر روی شئ مورد نظر کلیک کنیم و در پنجره ی properties بر روی event کلیک کرده و رویداد مورد نظر را انتخاب کرده و بر روی رویداد ایجاد شده در صفحه کد عملی که می خواهیم انجام شود را می نویسیم.

یکی از رویدادها که می تواند جنبه طراحی داشته باشد رویداد صفحه کلید است. برای کار با رویداد صفحه کلید باید در قسمت event بر روی keydown دابل کلیک می کنیم و در تکه کد ایجاد شده برای هر کدام از کلیدها که می خواهیم از آن استفاده کنیم شرط می گزاریم و کاری که می خواهیم را می نویسیم.

2-4پایگاه داده

من در #C با پایگاه داده SQL کار کردم

برای کار با SQL اول باید نصب آن را یاد بگیریم

دقت داشته باشید در ویندوزهای به جز Windows Server ، فقط نسخه Developer را می توانید به صورت کامل نصب کنید. کاربرد این نسخه بیشتر برای دانشجویان و برنامه نویسان می باشد. در این نسخه تمامی امکانات برنامه قابل دسترس می باشد. اگر قصد برنامه نویسی ندارید و فقط به قابلیت بانک اطلاعاتی SQL نیازمندید نسخه Standard Edition رانصبکنید . بهدلیلسادگینصبنسخهاستانداردازآموزشآنخودداریمیکنیمودراینمطلبفقطآموزشنسخه Developer را قرار می دهیم.

2-4-1نصب SQL

برای شروع نصب DVD را در داخل دستگاه قرار داده و مراحل زیر را انجام دهید .

نکته : قبل از شروع به نصب آنتی ویروس را غیر فعال کرده و اتصال خود به اینترنت و شبکه را قطع نمایید.

۱- ازداخلمحتویاتDVD پوشهیDeveloper را با کلیک کردن بر روی آن باز کنید.

۲- رویفایلSetup کلیک کنید تا مراحل نصب آغاز شود.

۳- چندلحظهصبر کنیدتاعملیاتادامهپیداکند.

۴- هماکنونصفحهایجدیدبازمیشود،ازسمتچپاینصفحهگزینهیInstallation را انتخاب کنید.

۵- حالازسمتراستصفحهایدرگزینه ۴ بازشدهبودگزینهی NewSQL Server stand-alone installation...راانتخابکنید.

۶- درصفحهیبعدOk را بزنید.

۷- صفحهایبانام Product Key باز می شود، در این صفحه گزینه ی Enter the product key را انتخاب و روی کلید Nextکلیک کنید.

۸- صفحهیجدیدیبانام License Terms باز می شود، در این صفحه گزینه ی I accept...راانتخابورویگزینهیNext کلیککنید.

۹- صفحهیبعدیکهبامیشود Setup Support Files نامدارد،درپایینصفحهبررویکلید Install کلیک کنید.

۱۰- نرمافزاردرحالنصبمیباشد،لطفاًچنددقیقهصبرکنید.

۱۱- پس از اتمام نصب صفحه ای با عنوان Setup Support Rules نمایان می شود، کلید Next را انتخاب کنید.

۱۲- صفحهیبعدیکهملاحظهمیکنید Feature Selection نام دارد، روی کلید Select All کلیک کنید تا تمام گزینه ها انتخاب شوند، حال برروی کلید Next کلیک کنید.

۱۳- هم اکنون صفحه ای با نام Instance Configuration باز می شود، گزینه ی Default Instance را انتخاب کرده و بر روی کلید Next کلیک کنید.

۱۴- درصفحهجدیدDisk Space Requirements بر روی کلید Next کلیک کنید.

۱۵- درصفحهیبعدیکه Server configuration نام دارد بر روی کلید Usethe same account کلیک کنید.

۱۶- پسازکلیکبررویکلیدمرحله ۱۵،پنجرهیکوچکینمایانمیشود . دراینپنجرهروبروی Account Name کلیککنیدو NT AUTHORITY\SYSTEM را انتخاب نمایید، هم اکنون روی کلید OK کلیک کنید.

۱۷- بهصفحهی Server configuration بازگردانده می شوید، حال بر روی کلید Next کلیک کنید.

۱۸- صفحهیبعدیکهبازمیشود Database Engine Configuration نام دارد، گزینه ی Mixed Mode را انتخاب کرده سپس کلید Add Current User را زده و بر روی کلید Next کلیک کنید.

۱۹- هماکنونصفحهایبان ام Analysis Services Configuration باز شده است، بر روی کلید Add Current User کلیک کرده و گزینه Next را بزنید.

۲۰- درقسمتبعدییعنی Reporting Services Configuration گزینه ی Install the native mode...راانتخابکردهوکلیدNext را کلیک کنید.

۲۱- درصفحه ی بعدی (Error And Usage Reporting) بدون انتخاب گزینه ای بر روی کلید Next کلیک کنید.

۲۲- حالصفحهیInstallation Rules باز شده است، دوباره بر روی کلید Next کلیک کنید.

۲۳- حالدرصفحهیبعدی (Ready to Install) کلید Install را بزنید.

۲۴- نصبنرمافزار آغاز شده است، این مرحله مدت زیادی طول خواهد کشید.

۲۵- درصفحهیبعدیتحتعنوان Installation Progress لیست قسمت های نصب شده را مشاهده می کنید، حال بر روی کلید Next کلیک کنید.

۲۶- هماکنونبهآخرینپنجره (Complete) رسیدیم. بر روی کلید Close کلیک کنید.

۲۷- حالمیتوانیدبایکبارRestart کردن سیستم خود از نرم افزارSQL2008 استفاده نمایید.

2-4-2پایگاه داده در سی شارپ

امروزه اکثر پایگاه داده های محبوب از نوع ***رابطه ای***(***Relational***) هستند. همانند پایگاه داده های عادی، در پایگاه داده های رابطه ای نیز دسترسی به اطلاعات ذخیره شده در جداول از طریق زبان پرس و جوی ساخت یافته یا همان ***SQL*** میسر میگردد که زبانی استاندارد است و توسط اکثر نرم افزارهای مرتبط با پایگاه داده مورد استفاده قرار می گیرد. از جمله سیستمهای پایگاه داده رابطه ای می توان به MS SQL Server، Oracle™، Sybase™، DB2™، Informix™ و MySQL™ اشاره کرد.

زبانهای برنامه نویسی از طریق یک ***Interface*** (یا همان نرم افزاری که ارتباط بین DBMS و یک برنامه را فراهم می کند) به پایگاه داده متصل شده و با آنها به تعامل میپردازند. در #C برقراری ارتباط با پایگاه داده از طریق ***ADO.Net*** انجام میشود.ADO.Net در حقیقت رابط بین نرم افزار و پایگاه داده است و امکانات ویژه ای را جهت دسترسی به اطلاعات موجود در آن در اختیار برنامه نویس قرار می دهد.

2-4-3مدل پایگاه داده رابطه ای (Relational Database Model)

هر ستون از جدول، فیلدی متفاوت را نشان میدهد. معمولا رکوردها در حدول منحصر بفرد هستند (بوسیله Primary Key) اما مقادیر فیلدهای مختلف می تواند مشابه با یکدیگر باشد. برای مثال، ۳ رکورد مختلف در فیلد Department از جدول Employee حاوی مقدار ۴۱۳ هستند.

Number	name	department	salary	Location
۲۳۶۰۳	میثم	۴۱۳	۱۰۰۰	تهران
۲۴۵۶۸	علی	۴۱۳	۱۵۰۰	تهران
۳۴۵۸۹	محمد	۶۴۲	۱۳۰۰	مشهد
۳۵۷۶۱	ناصر	۶۱۱	۱۸۰۰	اصفهان
۴۷۱۳۲	مریم	۴۱۳	۱۵۰۰	تهران
۷۸۳۲۱	فاطمه	۶۱۱	۲۵۰۰	اهواز

با توجه به اینکه حجم اطلاعات قابل ذخیره در یک جدول نامحدود است، از اینرو باید بتوان با استفاده از روشی تنها به آن قسمت از اطلاعات دسترسی پیدا کرد که مورد نظر کاربر است. برای این منظور از SQL استفاده می نماییم. SQL مجموعه دستوراتی را فراهم می نماید که با استفاده از آنها قادر خواهیک بود تا اطلاعات مورد نظر را از پایگاه داده انتخاب (SELECT) کرده و مورد استفاده قرار دهیم.

2-4-4نگاهی بر پایگاه داده رابطه ای

در ادامه مطالب این قسمت، با استفاده از یک پایگاه داده نمونه بنام Book با دستورات اولیه SQL آشنا خواهیم شد. پایگاه داده مورد نظر ما از چهار جدول تشکیل شده است. این جداول به ترتیب حاوی اطلاعاتی درباره نویسنده کتاب (Authors)، ناشر کتاب (Publishers)، کد شناسایی نویسنده (AuthorISBN) و عنوان کتاب (Titles) هستند. جدول Authors از سه فیلد تشکیل شده است که عبارتند از شماره اختصاصی هر نویسنده، نام و نام خانوادگی. در جدول زیر مشخصات جدول Authors نشان داده شده است.

نام فیلد	توضیحات
authorID	شماره شناسایی نویسنده را در پایگاه داده مشخص مینماید. در پایگاه داده Book این فیلد از نوع int تعریف شده و بصورت فیلدی اعلان شده که بطور خودکار مقدارش یک واحد یک واحد اضافه میشود (Auto-Increment Field). با اضافه شدن هر رکورد به پایگاه داده مقدار این فیلد یک واحد افزوده میشود که همین امر تضمین میکند که مقدار این فیلد همواره منحصر بفرد خواهد بود.
firstName	نام نویسنده کتاب. (از نوع string)
lastName	نام خانوادگی نویسنده. (از نوع string)

Authors		
lastName	firstName	authorID
Deitel	Harvey	1
Deitel	Paul	2
Nieto	Tem	3
Steinbuhler	Kate	4
Santry	Sean	5
Lin	Ted	6
Sadhu	Praveen	7
McPhie	David	8
Yaeger	Cheryl	9
Zlatkina	Marina	10
Wiedermann	Ben	11
Liperi	Jonathan	12
Listfield	Jeffrey	13

جدول Publisher از دو فیلد تشکیل گردیده است که نمایش دهنده شماره شناسایی ناشر و همچنین نام ناشر است. شکل زیر مشخصات جدول Publisher را نشان میدهد.

نام فیلد	توضیحات
publisherID	شماره شناسایی ناشر را در پایگاه داده نشان میدهد. این فیلد که از نوع int است و بصورت خودکار مقدارش افزوده میشود، فیلد Primary Key جدول Publisher نیز میباشد.
publisherName	نام ناشر کتاب. (از نوع string)

Publishers	
publisherName	publisherID
Prentice Hall	1
Prentice Hall PTG	2

جدول AuthorISBN شامل دو فیلد است که نشان دهنده شماره شناسایی نویسنده و شماره ISBN کتابهایی است که یک نویسنده خاص آنها را نوشته است. با استفاده از این جدول میتوان کتابهایی را که یک نویسنده خاص آنها را نوشته بدست آورد. شکل زیر مشخصات جدول AuthorISBN را نشان میدهد.

نام فیلد	توضیحات
authorID	شماره شناسایی نویسنده کتاب را نشان میدهد. با اشتفاده از این فیلد میتوان کتابهای مرتبط با هر نویسنده را در پایگاه داده پیدا کرد. این فیلد که از نوع int میباشد، در

	جدول Author نیز قرار دارد.
Isbn	شماره ISBN مربوط به هر کتاب. (string)

AuthorISBN	
authorID	ISBN
1	0130125075
2	0130125075
1	0130132497
2	0130132497
1	0130161438
2	0130161438
3	0130161438
1	0130284173
2	0130284173
3	0130284173
6	0130284173
7	0130284173
1	0130284181
2	0130284181
3	0130284181
8	0130284181
1	013028419X
2	013028419X
3	013028419X
1	0130293636
2	0130293636

جدول Titles از شش فیلد تشکیل شده است که عبارتند از ISBN مربوط به هر کتاب، عنوان کتاب، ویرایش، سال انتشار و شماره شناسایی ناشر، نام عکس جلد هر کتاب و قیمت هر کتاب. در شکل زیر مشخصات این جدول نشان داده شده است.

نام فیلد	توضیحات
Isbn	شماره ISBN مربوط به هر کتاب. (string)
Title	عنوان کتاب. (string)
editionNumber	شماره ویرایش کتاب (string)
Copyright	سال نشر کتاب (int)
publisherID	شماره شناسایی ناشر (از نوع int). مقادیر این فیلد میبایست با شماره ای در جدول Publisher همخوانی داشته باشد.
imageFile	نام فایلی که عکس روی جلد کتاب در آن قرار دارد . (string)
Price	قیمت کتاب. (از نوع اعداد حقیقی)

در شکل زیر رابطه بین جداول موجود در پایگاه داده Book نشان داده شده است. خط اول در هر جدول بیانگر نام جدول است. فیلدهایی که بصورت **Bold** نشان داده شده اند، بیانگر فیلدهای Primary Key هستند. همانطور که قبلا نیز گفته شد، فیلدهای Primary Key نمایش دهنده رکوردهایی منحصر بفرد در جدول هستند، از اینرو هر رکورد در جدول، میبایست دارای مقداری در فیلد Primary Key خود باشد و این مقدار میبایست منحصر بفرد باشد. از این اصل، بعنوان قانون "***جامعیت موجودیتها*** " یاد میشود (Rule Of Entity Integrity). توجه نمایید که در جدول AuthorISBN دو فیلد بعنوان Primary Key نشان داده است، که این بدین معناست که در این جدول Primary Key مرکب وجود دارد، از اینرو هر رکورد در این جدول میبایست دارای authorID و isbn منحصر بفردی (بطور مشترک) باشد. برای مثال، ممکن است رکوردهای بسیاری در جدول یافت شوند که مقدار فیلد authorID آنها برابر با ۲ باشد، اما چون ترکیب authorID و isbn میبایست منحصر بفرد باشد، از اینرو هیچ دو رکوردی نباید یافت شود که دارای authorID و isbn مشابه باشد. همچنین ممکن است رکوردهای مختلفی مقدار فیلد isbn آنها برابر با ۰۱۳۰۸۹۵۶۰۱ باشد، اما تنها یک فیلد یافت میشود که مقدار authorID آن برابر با ۲ و مقدار isbn آن برابر با ۰۱۳۰۸۹۵۶۰۱ باشد.

توجه : در صورتیکه مقداری برای فیلد Primary Key انتخاب نشود، اصل جامعیت موجودیتها نقض شده، از اینرو DBMS خطایی را گزارش میکند.

توجه : وارد کردن مقادیر مشابه برای فیلد Primary Key باعث می شود تا DBMS خطایی را گزارش نماید.

خطهای رسم شده بین جداول، رابطه (Relationship) بین جداول را نشان می دهد. برای مثال خط رسم شده بین جدول Publisher و Titles را در نظر بگیرید. همانطور که مشاهده میشود، در سمت Publisher، بر روی خط عدد ۱ قرار دارد و در سمت Titles علامت ∞ (بینهایت) قرار گرفته است. خط رسم شده بین این دو جدول، بیانگر یک رابطه "***یک به چند*** " (One-to-Many) است که بیان میدارد، به ازای هر ناشر در جدول Publishers، تعداد زیادی کتاب می تواند در جدول Titles قرار داشته باشد. همچنین توجه داشته باشید که خط رسم شده بین این دو جدول، از فیلد publisherID در جدول Publisher آغاز شده و به فیلد publisherID در جدول Titles ختم شده است. فیلد publisherID در جدول Titles یک Foreign Key است، بدین معنا که به ازای هر موجودیت در این جدول، یک مقدار منحصر بفرد در جدولی دیگر وجود دارد و این مقدار منحصر بفرد فیلد Primary Key جدول دوم است. Foreign Key در زمان ایجاد یک جدول مشخص می شود. با استفاده از Foreign Key، اصل "***جامعیت ارجاع***" (Rule Of Referential Integrity) مطرح می شود که بیان میدارد، مقدار هر Foreign Key باید در فیلد Primary Key جدولی دیگر وجود داشته باشد. با استفاده از Foreign Key میتوان اطلاعات موجود در جداول مختلف را با یکدیگر "***پیوند***" (join) زد و از آنها استفاده نمود. همواره رابطه ای "یک به چند" بین Primary Key و Foreign Key وجود دارد، بدین معنا که مقدار فیلد Foreign Key می تواند در جدول خودش چندین بار ظاهر شود اما در جدول دیگر تنها می تواند یکبار و آنهم بعنوان Primary Key ظاهر شود.

توجه : همانطور که گفته شد، همیشه رابطه ای "یک به چند " از سوی Primary Key به سمت Foreign Key وجود دارد.

توجه : استفاده از مقداری بعنوان فیلد Foreign Key که در Primary Key هیچ جدولی وجود نداشته باشد، اصل جامعیت ارجاع را نقض کرده، از اینرو DBMS خطایی را گزارش خواهد کرد.

5-4-2زبان پرس و جوی ساخت یافته (SQL)

در این قسمت، به بررسی مختصر زبان SQL خواهیم پرداخت. همانند زبانهای برنامه سازی، زبان SQL نیز از کلمات کلیدی و دستوراتی تشکیل شده است که دستورات کلیدی و مهم آنرا در جدول زیر مشاهده می کنید.

نام دستور SQL	توضیحات

SELECT	فیلدهای مورد نظر را از جدول یا جداولی انتخاب می کند.
FROM	جداولی را مشخص می کند که اطلاعات از آنها باید انتخاب شود و یا قرار است اطلاعات از آنها خدف گردد. در هر دستور SELECT و DELETE وجود دارد.
WHERE	شرایطی را مشخص می کند که تحت آن رکوردهایی خاص انتخاب خواهند شد.
INNER JOIN	رکوردهای متفاوتی را از جداول مختلف به یکدیگر متصل مینماید و مجموعه ای جدید از رکوردها را ایجاد مینماید.
GROUP BY	شرایطی را نشان میدهد که بوسیله آن دسته بندی اطلاعات انجام میشود.
ORDER BY	شرایطی را نشان میدهد که طی آن اطلاعات مرتب می شوند.
INSERT	داده را در جدولی خاص وارد می کند.
UPDATE	داده خاصی را در جدول مورد نظر بروز رسانی می کند. (تغییر میدهد)
DELETE	اطلاعات خاصی را از جدول مورد نظر حذف میکند.

دستور *SELECT*

یکی از ساده ترین دستورات در SQL، دستوری است که در آن اطلاعات از جدولی خاص انتخاب می شود. چنین عملی با استفاده از دستور SELECT صورت میگیرد و ساده ترین فرمت این دستور بصورت زیر است :

```
SELECT * FROM tableName
```

که در آن علامت "*" بیان میدارد که تمامی اطلاعات (تمامی ستونها) موجود در جدول انتخاب شده و به نمایش در خواهند آمد و *tableName* نام جدولی است که میخواهیم اطلاعات را از آن انتخاب نماییم.

```
SELECT * FROM Authors
```

برای انتخاب فیلد یا فیلدهایی خاص از یک جدول، میتوانیم نام فیلدهای مورد نظر آن جدول را در جلوی دستور SELECT و بجای * قرار دهیم و آنها را با کاما "," از یکدیگر جدا نماییم.

```
SELECT authorID, lastName FROM Authors
```

تنها فیلد authorID و lastName از جدول Authors را نشان میدهد.

نکته : در صورتیکه نام فیلد شامل فضای خالی (Space) باشد، در اینصورت میبایست نامو فیلد را داخل دو براکت "[]" قرار دهیم. (`SELECT [author ID] FROM Authors`)

دستور *WHERE*

در اکثر موارد کاربر در جدول به دنبال اطلاعاتی میگردد که دارای شرایط خاصی است. در این موارد تنها آن رکوردهایی که با شرایط مورد نظر کاربر همخوانی دارند میبایست نمایش داده شوند. SQL با استفاده از دستور WHERE در دستور SELECT، شرایط مورد نظر را اعمال می نماید. ساده ترین فرم دستور SELECT که بهمراه WHERE باشد، بصورت زیر است :

```
SELECT fieldName1,fieldName2,… FROM tableName
WHERE criteria
```

که در آن *fieldName*، نام فیلدهای مورد نظر، *tableName* نام جدولی که اطلاعات از آن استخراج می شود و *criteria* شرایطی است که بر اساس آن جستجو در جدول صورت میپذیرد. برای مقال، درصورتیکه بخواهیم عناوین کتابهایی را بیابیم که تاریخ نشر آنها بعد از سال ۱۹۹۹ است، از دستور SELECT زیر استفاده می کنیم :

```
SELECT title
FROM Titles
WHERE copyright > 1999
```

نکته : در زبان #C برای اجرای دستورات SQL از یک رشته استفاده می کنیم که این رشته حاوی کل دستور (Query) مورد نظر ما است. همچنین برای بالا رفتن خوانایی برنامه، دستورات SQL را در خطوط جدا مینویسیم.

دستور WHERE می تواند خاوی عملگرهای <، >، =<، =>، =، <> و عملگر LIKE باشد. عملگر LIKE برای "***تطبیق الگو***" (Pattern Matching) مورد استفاده قرار میگیرد و بهمراه * و ? مورد استفاده قرار میگیرد. با استفاده از تطبیق الگو، میتوان به دنبال رشته ای گشت که حاوی الگوی مورد نظر است. برای مثال، query زیر، تمامی رکوردهایی از جدول Authors را نشان میدهد که lastName آنهابا کاراکتر "D" شروع شده باشند :

```
SELECT * FROM Authors

WHERE lastName LIKE 'D*'
```

توجه نمایید، استفاده از کاراکتر * بیان میدارد که برای تطبیق الگو تنها کافیست تا کاراکتر اول مقدار D داشته باشد و سایر کاراکترها هر مقداری می توانند داشته باشند. همچنین اهمیتی ندارد که چه تعدا کاراکتر بعد از کاراکتر D در رشته وجود دارد و تنها اهمیت برای ما وجود کاراکتر D در ابتدای رشته است.

نکته : تمامی سیستم های پایگاه داده از عبارت LIKE پشتیبانی نمیکنند.

نکته : در اکثر سیستمهای پایگاه داده، به جای استفاده از کاراکتر "*" در عبارت LIKE، از کاراکتر "%" استفاده میشود.

نکته : در برخی از سیستمهای پایگاه داده، کاراکترهای رشته، Case Sensitive هستند.

نکته : بهتر است برای تمیز دادن دستورات و عبارات SQL، آنها را با حروف بزگ بنویسیم.

توجه نمایید در رشته ای که مورد تطبیق الگو قرار میگیرد، درصورتیکه کاراکتر "?" قرار گیرد، بدین معناست که بجای کاراکتر "?" تنها یک کاراکتر قرار خواهد گرفت. برای مثال در query زیر، تمامی رکوردهایی از جدول Authors نمایش داده می شوند که lastName آنها با هر کاراکتری شروع شده باشد و به دنبال آن کاراکتر "i" آمده باشد و بعد از آن هر تعداد کارارکتر قرار داشته باشد :

```
SELECT * FROM Authors
WHERE lastName LIKE '?i*'
```

نکته : اکثر پایگاه های داده، به جای استفاده از کاراکتر "?" از کاراکتر "_" در عبارت LIKE استفاده میکنند.

نکته : توجه نمایید که پس از عبارت LIKE، رشته مورد نظر درون یک جفت '' قرار میگیرد.

دستور ORDER BY

نتیجه یک query می تواند بصورت صعودی یا نزولی مرتب گردد. این کار با استفاده از دستور ORDER BY انجام میشود. ساده ترین فرم این دستور بصورت زیر است :

```
SELECT fieldName1, fieldName2, … FROM tableName
ORDER BY field ASC
```

```
SELECT fieldName1, fieldName2, … FROM tableName
ORDER BY field DESC
```

که در *field*، نام فیلدی است که نتایج query بر اساس آن مرتب می شود. در صورتیکه بخواهیم نتایج بصورت صعودی مرتب شوند از کلمه ASC و درصورتیکه بخواهیم نتایج بصورت نزولی مرتب شوند از کلمه DESC استفاده می نماییم. بطور پیش فرض، نتایج query بصورت صعودی نمایش داده میشوند.

```
SELECT * FROM Authors
ORDER BY lastName DESC
```

همچنین، با استفاده از ORDER BY میتوان نتایج query را بر اساس چند فیلد مختلف نیز مرتب نمود :

```
ORDER BY field1sortingOrder, field2sortingOrder, …
```

که در آن *field*، فیلدهایی هستند که مرتب سازی بر اساس آنها صورت میگیرد و *sortingOrder* نوع مرتب سازی هر فیلد را نشان میدهد. توجه نمایید، در مواردی که دو یا چند فیلد برای مرتب سازی در نظر گرفته شده اند، ابتدا مرتب سازی بر اساس فیلد اول انجام می شود و پس از آن مرتب سازی بر اساس فیلدهای دیگر به پیش میرود. در مثال زیر،ابتدا نتایج query بر اساس نام خانوادگی (lastName) و سپس بر اساس نام (firstName) مرتب میشوند :

```
SELECT * FROM Authors
ORDER BY lastName, firstName
```

توجه نمایید که در یک query می توان از ترکیب دستورات WHERE و ORDER BY استفاده نمود :

```
SELECT isbn, title, price FROM Titles
WHERE title LIKE '%How To Program'
ORDER BY title DESC
```

2-4-6 ادغام داده ها از جداول مختلف : *INNER JOIN*

طراحان پایگاه داده، معمولا اطلاعات را به جداول مختلفی تقسیم میکنند تا مطمئن باشند که در پایگاه داده اطلاعات بیهوده و زائد ذخیره نمیشود. برای مثال، پایگاه داده Book شامل دو جدول Authors و Titles است. ما با استفاده از جدول AuthorISBN، لینکی بین نویسنده های مختلف و عناوین کتبی که نوشته اند برقرار میکنیم. در صورتیکه این اطلاعات را به جداول مختلف تقسیم نمیکردیم مجبور بودیم تا اطلاعات نویسنده را برای هر عنوان کتاب در جدول Titles در نظر بگیریم و از اینرو یکسری اطلاعات اضافی و تکراری در جدول ذخیره میشد چراکه برای عناوین کتبی که نویسنده آنها یکسان است، اطلاعات نویسنده بطور تکراری ذخیره میگردید.

همچنین برای مقاصد آنالیز و تجزیه و تحلیل، ضروری است تا اطلاعات مختلف از جداول مختلف با یکدیگر ادغام شوند و یک مجموعه اطلاعاتی جدید را ایجاد نمایند. با استفاده از عملی تحت عنوان "***ادغام جداول***"، این عمل قابل اجرا است و بوسیله INNER JOIN در دستور SELECT آنرا در SQL عملی میکنیم. یک INNER JOIN رکوردهای مختلفی را از جداول متفاوت با یکدیگر ادغام میکند و این عمل را از طریق تست کردن مقادیر فیلدهایی انجام میدهد که در جداول مورد نظر عمومیت دارند. ساده ترین فرم INNER JOIN بشکل زیر است :

```
SELECT fieldName1, fieldName2, …
FROM table1
INNER JOIN table2
ON table1.fieldName = table2.fieldName
```

که در آن قسمت بعد از عبارت ON، فیلدهایی را نشان میدهد از دو جدول با یکدیگر مقایسه میشوند تا معین شود چه رکوردهایی با یکدیگر ادغام میشوند. برای مثال، query زیر لیستی از تمام نویسنده ها به همراه ISBN مربوط به کتابهای نوشته شده توسط آنها را ایجاد میکند :

```
SELECT firstName, lastName, isbn
FROM Authors
INNER JOIN AuthorISBN
ON Authors.authorID = AuthorISBN.authorID
ORDER BY lastName, firstName
```

این query فیلدهای firstName و lastName از جدول Authors را با فیلد isbn از جدول AuthorISBN ادغام میکند و نتیجه را بر اساس فیلدهای lastName و firstName بصورت صعودی مرتب مینماید. در این query به فرمت عبارت نوشته شده بعد از عبارت ON که بصورت

*tableName.fieldName*است توجه نمایید. در این قسمت مشخص میشود که عمل ادغام بین دو جدول از طریق کدام فیلد انجام خواهد گرفت. استفاده از فرمت نوشتاری "*tableName.*" در مواردی ضروری است که نام فیلد در هر دو جدول یکسان باشد.

ادغام اطلاعات از جداول Authors، AuthorISBN، Titles و Publisher

در اینجا می خواهیم برای پایگاه داده Book یک query تعریف کنیم که توسط آن عنوان کتاب، شماره ISBN، نام نویسنده، نام خانوادگی نویسنده، سال انشار کتاب و نام ناشر هر کتاب نمایش داده شود. برای کتابهایی که بیش از یک نویسنده دارند، این query میبایست رکوردهای مجزایی را برای هر نویسنده نمایش دهد. توجه نمایید که در این query نیاز است تا بین هر چهار جدول ادغام صورت گیرد. این query در زیر نمایش داده شده است :

```
SELECT     Titles.Title,     Titles.ISBN,     Authors.FirstName,
Authors.LastName, Publishers.PublisherName

FROM

      (

       Publishers

       INNER JOIN Titles

       ON Publishers.PublisherID = Titles.PublisherID

      )

INNER JOIN

      (

       Authors

       INNER JOIN AuthorISBN

       ON Authors.AuthorID = AuthorISBN.AuthorID
```

```
)

ON Titles.ISBN = AuthorISBN.ISBN
ORDER BY Titles.Title;
```

حال به بررسی این query میپردازیم. در خطوط ۱و۲ این query، فیلدهای مورد نظر که بعنوان نتیجه در خروجی نمایش داده میشوند، قرار گرفته اند. توجه نمایید که ترتیب قرار گرفتن فیلدها در این لیست، از این جهت اهیمت می یابد که فیلدها در خروجی به همین ترتیب نمایش داده خواهند شد. این query، فیلدهای title و isbn از جدول Titles، فیلدهای lastName و firstName از جدول Authors، فیلد copyright از جدول Titles و فیلد publisherName از جدول Publishers را انتخاب مینماید. برای اینکه متن این query خوانا تر باشد، قبل از نام هر فیلد نام جدول آنرا نیز آورده ایم. (به این روش Fully Qualified Name گفته میشود.) در خطهای بعدی، عمل ادغام بین جداول مختلف صورت گرفته است. در این query از سه عمل INNER JOIN استفاده شده است. باید به نکته توجه شود، اگرچه INNER JOIN بر روی دو جدول انجام میشود، اما نتیجه آن می تواند بعنوان ورودی برای INNER JOIN دیگری مورد استفاده قرار گیرد چراکه نتیجه یک INNER JOIN خود یک جدول است. همچنین برای اولویت بخشیدن به اجرای دستورات از پرانتز استفاده نموده ایم. در ابتدا با دستور زیر آغاز میکنیم :

```
(Publishers INNER JOIN Titles
  ON Publishers.publisherID = Titles.publisherID)
```

که جدول Publishers و Titles را با استفاده از شرایطی که در آن فیلدها publisherID دو جدول با یکدیگر منطبق هستند، ادغام میکند. جدول موقتی که از انجام این INNER JOIN تشکیل میشود حاوی اطلاعات کتاب و ناشر مربوط به آن است.

دیگر INNET JOIN این query عبارت است از :

```
( Authors
 INNER JOIN AuthorISBN
 ON Authors.AuthorID = AuthorISBN.AuthorID )
```

که جدول Authors و AuthorISBN را در شرایطی که فیلد AuthorID با یکدیگر منطبق باشند، ادغام مینماید.

سومین INNER JOIN در این query عبارت است از :

```
(
 Publishers
 INNER JOIN Titles
 ON Publishers.PublisherID = Titles.PublisherID
)
INNER JOIN
(
 Authors
 INNER JOIN AuthorISBN
 ON Authors.AuthorID = AuthorISBN.AuthorID
)

ON Titles.ISBN = AuthorISBN.ISBN
```

که باعث ادغام دو جدول موقتی ایجاد شده از دو INNER JOIN قبلی، تحت شرایطی میشود که در آن فیلد Titles.isbn برای هر رکورد در اولین جدول موقت با فیلد AuthorISBN.isbn برای هر رکورد از جدول

موقت دوم، منطبق باشند. نتیجه نهایی این INNER JOIN نیز، جدول موقتی است که نتیجه مورد نظر را در بر دارد.

در انتهای query نیز، نتایج بر اساس فیلد Titles.title مرتب میشوند.

دستور *INSERT*

این دستور باعث وارد کردن رکوردی جدید در جدول میشود. ساده ترین فرم این دستور بشکل زیر است :

```
INSERT INTO tableName (fieldName1, fieldName2, … )
VALUES (value1, value2, …)
```

که در آن *tableName* جدولی است که در آن رکورد جدید وارد میشود. به دنبال *tableName* لیستی از فیلدها قرار میگیرد و پس از آن عبارت VALUES و به دنبال آن لیستی از مقادیر. توجه نمایید که به ازای هر *fieldName*، مقدار *value* در جدول وارد میشود. همچنین باید توجه شود که ترتیب و نوع مقادیر وارد شده در *value* میبایست با نام و نوع *field* مورد نظر همخوانی کامل داشته باشد. برای مثال، اگر *fieldName1* بعنوان firstName در نظر گرفته شود، آنگاه مقدار *value1* باید رشته ای باشد که درون یک جفت

' ' قرار گرفته و بیانگر نام باشد. بعنوان یک مثال، query زیر را در نظر بگیرید :

```
INSER INTO Authors (lastName, firstName)
VALUES ('Smith' , 'Sue')
```

این query باعث میشود تا رکوردی جدید در جدول Authors وارد شود. توجه نمایید در دستور INSERT حتما برای فیلدهای Primary Key میبایست مقداری وارد گردد، اما در اینجا چون فیلد Primary Key، فیلد authorID است و این فیلد از نوع auto-increment تعریف شده، نیازی به مقداردهی ندارد، چراکه با وارد شده هر رکرود جدید بطور خودکار مقداری برای آن در نظر گرفته میشود. همچنین دقت کنید، بغیر از فیلدهای Primary Key، در صورتیکه برای فیلدی از یک جدول در دستور INSERT مقداری در نظر گرفته نشود، مقدار آن فیلد تهی (Null) در نظر گرفته خواهد شد.

نکته : SQL برای نشان دادن رشته ها از یک جفت ' ' استفاده میکند. برای نشان دادن رشته هایی که خود حاوی ' هستند (مانند O'Reily) باید از فرمت 'O' 'Reily' استفاده شود.

دستور *UPDATE*

این دستور باعث ایجاد تغییر در داده های جدول میشود و ساده ترین فرم آن بصورت زیر است :

```
UPDATE tableName
SET fieldName1 = value1, fieldName2 = value2, …
WHERE criteria
```

که در آن *fieldName* هایی که مقادیر آنها تغییر مییابند مشخص شده و مقادیر جدید (*value*) به آنها تخصیص داده میشود. وجود عبارت WHERE در دستور UPDATE باعث میشود تا رکوردهایی که این فیلدها در آنها تغییر می یابند معین شوند. برای مثال :

```
UPDATE Authors
SET lastName = 'Jones'
```

```
WHERE lastName = 'Smith' AND firstName = 'Sue'
```

باعث میشود تا رکوردی که در آن Sue Smith قرار داشته است، به Sue Jones تغییر یابد.

توجه : عدم استفاده از عبارت WHERE پس از دستور UPDATE ممکن است باعث بروز خطایی منطقی گردد.

دستور *DELETE*

دستور DELETE باعث خذف داده ها از جدول میشود و ساده ترین فرم آن بصورت زیر است :

```
DELETE FROM tableName
WHERE criteria
```

که در آن criteria مشخص کننده رکورد یا رکوردهایی است که از جدول حذف میشوند.

```
DELETE FROM Authors
WHERE lastName = 'Jones' AND firstName = 'Sue'
```

باعث خذف شدن رکوردی میشود که نام و نام خانوادگی آن Sue Jones است.

نکته : عبارت WHERE ممکن است با چندین رکورد در جدول مورد نظر همخوانی داشته باشد، از اینرو در دستور DELETE دقت نمایید که شرط نوشته شده در WHERE تنها با رکورد مورد نظر شما همخوانی داشته باشد تا از خذف شدن ناخواسته اطلاعات جلوگیری شود.

2-5مدل شیء ADO.NET(ADO.Net Object Model)

مدل شیء ADO.Net فراهم کننده API ای است که بوسیله آن می توان از طریق کد نویسی به پایگاه داده دسترسی پیدا کرد. ADO.Net مخصوص Net Framework. طراحی شده و نسل جدید ADO میباشد. (کلمه ADO مخفف ActiveX Data Objects است.)

در Net Framework.، System.Data فضایی است که در آن تمامی API های مربوط به ADO.Net قرار دارند. Namespace های پایه ADO.Net عبارتند از System.Data.OleDb و System.Data.SqlClient که با استفاده از آنها امکان دسترسی و ایجاد تغییر در Data Source های مختلف فراهم میگردد. (منظور از Data Source هر منبعی داده ایست که میتوان در آن اطلاعات و داده ها را ذخیره نمود.)

System.Data.OleDb شامل کلاسهایی است که بوسیله آنها می توان به هر Data Source ای متصل شد، در حالیکه System.Data.SqlClient تنها شامل کلاسهایی جهت استفاده از MS SQL Server 2000 است.

شروع برنامه نویسی با ADO.Net– اتصال به MS Access

برای آغاز کار برنامه نویسی و اتصال به پایگاه داده، با پایگاه داده Access شروع میکنیم که پایگاه داده ای ساده بوده و به همراه بسته نرم افزاری MS Office قابل نصب است. توجه کنید در برنامه زیر، از ساده ترین روش برای اتصال به پایگاه داده استفاده شده و مطالب پیشرفته تر را در آینده بررسی خواهیم نمود.

برای اتصال به پایگاه داده Access میبایست از کلاسهای موجود در System.Data.OleDbاستفاده نماییم.

مراحل کار با پایگاه داده بشرح زیر میباشد :

- اتصال به پایگاه داده
- بدست آوردن اطلاعات مورد نظر از پایگاه داده
- نمایش اطلاعات بدست آمده
- قطع ارتباط با پایگاه داده

برای اتصال به پایگاه داده Access از شیء OleDbConnection استفاده میکنیم. توجه کنید که برای اتصال به هر پایگاه داده ای نیاز به یکسری پارامترها و مسیرها است که این اطلاعات در رشته ای تحت عنوان connection string قرار داده میشود و از طریق اطلاعات این رشته به پایگاه داده متصل میشویم. برای اتصال به Access نیز میبایست از یک connection string استفاده نماییم. برای مثال، رشته زیر را در نظر بگیرید :

```
string connectionString =
  "provider=Microsoft.Jet.OLEDB.4.0;" +
  "data source=F:\\Samples\\Books.mdb";
```

این رشته، نمونه ای از یک connection string جهت برقراری ارتباط با پایگاه داده Access است. در قسمت provider نام شرکت پشتیبانی کننده پایگاه داده نوشته میشود که برای Access این مقدار عبارت است از Microsoft.Jet.OLEDB.4.0 و در قسمت data source محل قرار گرفتن پایگاه داده مشخص میگردد. (محل فیزیکی که در آن پایگاه داده ذخیره شده است.)

همچنین به استفاده از "\\" جهت نمایش "\" توجه نمایید.

در مثال زیر، برنامه نمونه ای مطرح شده که در آن نحوه برقراری ارتباط با پایگاه داده Access از طریق شیء OledbConnection به نمایش در آمده است.

```
using System;
using System.Data;
using System.Data.OleDb;

classOleDbConnectionAccess
{
  publicstaticvoid Main()
  {
    // بیان رشته ای که در آن جزئیات ارتباط با پایگاه داده مشخص میشود.
    string connectionString = "provider=Microsoft.Jet.OLEDB.4.0;" +
                              "data source=F:\\Samples\\Books.mdb";

    // ایجاد شیء جدیدی از OleDbConnection جهت برقراری ارتباط با پایگاه
    // که connectionStringبه سازنده آن ارسال میشود.
    OleDbConnection myOleDbConnection =
            newOleDbConnection(connectionString);

    // OleDbCommand ایجاد شیء
```

```
OleDbCommand myOleDbCommand = myOleDbConnection.CreateCommand();


// SQL قرار دادن ویژگی CommandText از شیء OleDbCommand به دستور
// که توسط آن یک سطر از جدول Authors انتخاب میشود.//
myOleDbCommand.CommandText =
  "SELECT * "+
  "FROM Authors "+
  "WHERE authorID=1";


// باز کردن ارتباط بین پایگاه داده با استفاده از متد Open()
myOleDbConnection.Open();


// ایجاد شیء OleDbDataReader و فراخوانی متد ExecuteReader() از
// شیء OleDbCommand جهت اجرای دستور SELECT
  OleDbDataReader myOleDbDataReader =
                      myOleDbCommand.ExecuteReader();


// خواندن یک سطر از OleDbDataReader
// با استفاده از متد Read()
myOleDbDataReader.Read();


// نمایش نتیجه و خروجی
Console.WriteLine("myOleDbDataReader[\" firstName\"] = "+
  myOleDbDataReader["firstName"]);
```

```
    Console.WriteLine("myOleDbDataReader[\" lastName\"] = "+
      myOleDbDataReader["lastName"]);
    Console.WriteLine("myOleDbDataReader[\" AuthorID\"] = "+
      myOleDbDataReader["authorID"]);

    // Close() بستن شیء OleDbDataReader با استفاده از متد
    myOleDbDataReader.Close();

    // OleDbConnection بستن ارتباط بین پایگاه داده از شیء
    myOleDbConnection.Close();
  }
}
```

خروجی این برنامه بشکل زیر است :

```
myOleDbDataReader[" firstName"] = Harvey
myOleDbDataReader[" lastName"] = Deitel
myOleDbDataReader[" AuthorID"] = 1
```

توجه نمایید که در اینجا تنها یک سطر از پایگاه داده فراخوانی شده و با تغییر دستور SQL میتوان نتیجه های دلخواه را بدست آورد.

2-6نوشتن کلاسها یکی دیگر از کارهایی است که انجام داده ام

کلاسهاواشیاء

مادنیارامرکبازاشیاءدرکمیکنیم .

شماچیزیجزقطعاتپلاستیکیوشیشهایادغامشدهبامحیطرانمیبینید،شمابطورطبیعیاشیاءمتمایزرامیبینید .

یککامپیوتر،یکصفحهکلید،یکمانیتور،اسپیکرها،مدادوکاغذ .

مطلبمهماینکه،قبلازاینکهخودتانتصمیمبگیرید،ایناشیاءرابهگروهمبدلکردهاید .

شماکامپیوتررویمیزتانرابهعنوانیکنمونهیخاصازیکنوعگروهبندیمیکنید .اینکامپیوتریکنمونهازنوعکامپیوتراست.

نظریهپیشتپردهبرایبرنامهسازیشیگرا،مدلسازیصحیحدنیابرایبرنامههایکامپیوتریاست .

اینبرنامههابایدتمایلانسانرادرموردنمایشتکتکاشیاءونوعآنهامنعکسسازد . درC#اینکاررابااایجادیک

کلاسجهتتعریفیکنوعدادهوایجادیکنمونهازآنکلاسبرایمدلکردنچیزیانجاممیدهید.یک کلاس یک نوع داد ه ی جدید را تعریف می کند. هر کلاسی خصوصیات مشترک هر شی از آن نوع جدید را تعریفمی کند.

برای مثال، کلاس Carرا در نظر بگیری د . ماشین من و شما هر دو به کلاس تعلق دارند . Car آنها از نوع داده ی Car هستند .یک شی، یک نمون ه ی منحصر به فرد از یک کلاس است . هر ماشین منحصربه فرد، یک نمونه از کلاس car است . پس یک شی است و شی نیز یک چیز است.

تعریفیککلاس:

در زمان تعر یف یک کلاس، ویژگی ها و رفتار اشیاء آن نوع داده را تعریف می کنید. در c# ویژگی ها را با فیلد عضو شرح می ده ی.

```
class Dog
{
private int weight; // member field

private String name; // member field
```

فیلدهای کلاس برای نگهداری حالت شی استفاده می شوند. برای مثال، حالت Dog با وزن و نام جاری آن تعریف می شود.حالت یک کارمند با حقوق و سطح مدیریت و نرخ کارائی آن تعریف می شود. شما رفتار نوع داد هی جدیدتان را با متدهاتعریف می کنید. متدها کدی را برای انجام یک عمل در بر می گیرند.

```
class Dog
{
```

```
private int weight;
private String name;
public void Bark( ) // member method
{
// code here to bark

}
```

کلمات کلیدی public و private معرف های دسترسی هستند که سطح دسترسی به آن فیلد یا متد را مشخص می کنند. اعضای private فقط در داخل همان کلاس قابل دسترسی هستند، ولی اعضای public از بیرو ن کلاس نیز قابل دسترسی خواهند بود.

یک کلاس تعدادی متد برای کار با آن کلاس تعریف م ی کند. کلاس Dog احتمالا متدهایی برای پارس کردن و خوردن و چرت زدن ۲ و... دارد. کلاس کارمند متدهایی برای تسویه حقوق، بازدیدهای سالیانه و ارزیابی اهداف کارائی در بر می گیرد.

متدها با تغییر دادن مقادیر فیلدهای عضو م ی توانند حالت شی را دستکاری کنن د . یک متد م ی تواند با اشیاء دیگر از همان نوع یا انواع دیگر تعامل داشته باشن د . تعامل بین اشیاء برای برنام ه نویسی شی گرا بسیار مهم است . برای مثال، یک متد در کلاس سگ، حالت آن را تغییر می دهد. (یک متد feed() ن سگ را تغییر می دهد). با سگهای دیگر تعامل دارد ،Bark)و (Sniff و یا با انسان تعامل دارد .(BegForFeed) ممکن است یک شی کالا با شی مشتری تعامل داشته باشد.

در یک برنامه ی شی گرای خوب، شما اشیائی را طرح م ی کنید که در دامنه مسئله شما چیزهائی ر ا نشان م ی دهند. سپس کاربرنامه را با تخصیص مسئولیت به اشیاء (بر اساس توانایی آنها)، مابین اشیاء تقسیم خواهید کرد.

روابطکلاس:

برقراریروابطمابینکلاسها،قلبطراحیشیگرااست . کلاسهاباروشهایمختلفیبابقیهدرتعاملبودهومرتبطهستند . سادهترینتعاملزمانیاستکهمتدیازیککلاس،متدکلاسدیگریرافراخوانیمیکند .برایمثال،کلاس Manager متد UpdateSalary یکشیازکلاسکارمندرافراخوانیمیکند .
میگوئیمکلاس Manager و کلاس Employee انجمنهستند .انجمنمابینکلاسهابهمعنیتعاملمابینآنهاست.

بعضیازانواعدادهاپیچیدهازانواعدیگریمرکبمیشوند .برایمثال،یکخودرو،ازچرخها،موتور،سیستمانتقالو ...تشکیلشدهاست . برایمدلسازیخودرو،یککلاسچرخ،یککلاسموتورویککلاسانتقالایجادمیشود .
پسمیتوانیدکلاسماشینراایجادکنید . هرماشینچهارنمونهازکلاسچرخ،یکنمونهازکلاسموتورویکنمونهازکلاسانتقال دارد . این ترکیب عموما رابطه Has-a خوانده می شو د . روش دیگر نمایش این رابطه آن است که کلاس خودرو، کلاس چرخ، موتور و انتقال را تجمع می کند. یا کلاس کار از اشیاء چرخ، موتور و انتقال تشکیل می شود. این رابطه را Is-a گویند.
در بعضی از زبان ها همچون c++ مابین روابط is-a و has-a تفاوت وجود دارد، ولی در C# این تفاوت اعمال نمی شود.
رابطه ی تجمع به شما اجازه می دهد کلاس های پیچیده را با اسمبل کردن و ترکیب کلاس های ساده ایجاد کنید . چارچوب NET. یک کلاس Stringبرای بکار بردن رشته های متنی فراهم می کند. ممکن است کلاس Address خود را با پنج رشته متنی ایجاد کنید. سپس کلاس Employee را طوری ایجاد کنید که یک عضو آن از نوع Address است.

از جمله کارهای دیگری که انجام داده ام طراحی صفحات وب با ASP بوده است که کار با ASP در فصل سوم بطور مفصل توضیح داده شده است.

فصل سوم
ASP

مقدمه

افراد به یکی از این دو دلیل، تصمیم بهیادگیری تکنولوژی جدید کامپیوتری
می گیرند:یا نیاز دارند یا مجبور هستند.
در دنیای امروز کامپیوترها مخصوصا وقتی پای اینترنت به میان کشیده می شود، هیچ کس همه چیز را نمی داند، بهتر است به این امید باشید که هر وقت لازم باشد ،
می توانید آنرا یاد بگیرید. به طور کلی هیچ فردی نیست که به زبان خاصی برنامه نویسی کند. در یک پروژه مجبور به یادگیری چند یک یا زبان یا تکنولوژی کامپیوتری نشود و خط مستقیمی را طی کند.
اخیرا همین اتفاق در مورد در یک پروژه Active Server Page افتاد . قبل از آن هیچ تلاشی برای یادگیری ASP نکرده بودم . از روی اینترنت منابعی را بررسی کردم. ولی اصل موضوع کامل نبود . تنها روشی که من را به سمت یادگیری پیش برد ساختن پروژه به طور عملی بود. بهتر است برای فرا گرفتن مطلبی که درباره اش چیزی نمی دانیم یک پروژه عملی بسازیم و کار را از ابتدا تا انتها دنبال کنیم ، در اغلب اوقات باعث می شود جزئیات مطلب را نیز یاد بگیریم.

3-1 ASPچیست؟

یک اشتباه مشترک در میان بعضی از تولید کنندگان وب اینست که ASP یک زبان برنامه نویسی است ولی این طور نیست .در واقع ASP محیطی برای برنامه نویسی است.
به طوری که می توان برای ساختن صفحات ASP خود از هر زبانی که وب سرور می فهمد استفاده کرد.دو زبان معمول و رایجتر VBScript (زیر مجموعه ای از ویژوال بیسیک) و Jscript هستند.
Personal web server هر دو زبان را پشتیبانی می کند.رایجترین زبان برای تدوین و کنار هم گذاشتن ASP ها VBScript است.

به کار بردن یک زبان

```
<%@ language="VBSCRIPT" %>
<HTML>
<HEAD>
<TITLE> Primery outpost </TITLE>
<HEAD>
```

افزودن متن با استفاده از ASP

```
</HEAD>
<BODY bgcolor="32c8oo">
<IMG SRC="IMAGES/logo.gif "ALT"primery outpost logo"
HEIGHT="67" WIDTH="496">
<P>
```

3-2اشکال زدائی کد

هر چقدر هم با برنامه ریزی پیش روید همیشه امکان بروز خطا وجود دارد . ممکن است خطاهایتایپی وجود داشته باشد مثل جا گذاشتن یک نقل قولو ... گاهی اوقات پیدا کردن خطا کاری مشکل است .
اگر مستقیما روی سرور کار می کنیم, از IIS استفاده می کنیم و از مزایای Microsoft Script Debugger بهره مند می شویم با این برنامه می توانیم نقاط انفصال را روی در اسکریپت طوری تنظیم نمائیم که در هر لحظه از یک خط عبور نمائیم و مقادیر و متغیرها و چگونگی تغییر متغیرها در هنگام اجرای اسکریپت را نشان دهد .

در استفاده از Microsoft Script Debugger برای تجزیه و تحلیل صفحات خود ابتدا باید server side debugger را روی سرور فعال نماییم. در Internet services Mannager روی پیش فرض web siteکلیک می کنیم و properties را انتخاب می نمائیم روی home directory و سپس configuration کلیک می کنیم و سپس Enable ASP server side debugger را فعال می کنیم. اشکال زدا هر وقت که به یک خطا یا نقطه توقف داخلی برسد کارش را شروع می کند و فایلی را باز می کند و منابع خطا را نشان می دهد.

یک اشکال زدای خوب برای خطا یابی می تواند در ساعتها وقت صرفه جوئی کند .

3-3واسط کاربری(user interface)

3-4تعریف چند اصطلاح

- واسط کاربر : به اشکال گرافیکی و متنی مربوط می شود که در داخل یک برنامه کاربردی یا سایت وب به هدایت کاربر کمک می کند.
- واسط گرافیکی کاربر (Graphical user interface): شامل گرافیک ها و دکمه ها و اجزای دیگری است که بکار گیری سیستم را برای کاربر آسانتر می کند.

افزودن واسط کاربری

```
<%@ language="VBSCRIPT" %>
<HTML>
<HEAD>
<TITLE> Primery outpost </TITLE>
</HEAD>
<BODY bgcolor="9caboo">
<IMG SRC="IMAGES/logo.gif "ALT"primery outpost logo"
HEIGHT="67" WIDTH="496">
</A>
<A href ="news/news.asp">
<IMG SRC="IMAGES /archives.gif border =0   ALINE= "middle"   ALT= "Archied newes'
HEIGHT="39"   WIDTH="39">
<A/>
</BODY>
</HTML>
```

3-5گذاشتن تصویر در زیر تصویر

```
<IMG SRC="IMAGES /archives.gif border =0   ALINE= "middle"   ALT= "Archied newes'
HEIGHT="39"   WIDTH="39">
<A/>
<BR CLEAR =”LEFT”>
OUR NEWES WILL GO HERE.
```

این کد به مرورگر می گوید که بعد از این که سمت چپ تصاویر یا اقلام تصاویر روشن شد به خط بعدی برود.

3-6افزودن یک LINK برگشتی به صفحه خانگی

```
<BODY bgcolor="9caboo">
<A Href=http://localhost/hndex.asp”>
<IMG SRC="IMAGES/logo.gif "ALT"primery outpost logo"
HEIGHT="67" WIDTH="496">  ALIGN=”left” 9a:border=0>
</A>
<A href=”newes.asp”>
```

3-7بانک اطلاعاتیبا استفاده از ASP

آنچه که در این مبحث بیان می گردد:

- ایجادبانک اطلاعاتی و ODBC DSN
- **درج رکوردها در بانک اطلاعاتی**
- **ایجاداتصال به بانک اطلاعاتی**

- مفاهیم مقدماتی SQL : درج کردن (Insert)
- افزودن خصوصیات محاوره ای (Interactive) به فرم ها
- مدل سازی داده ها (Data Modeling)
- رسیدگی به خطاها

- تکمیل فرم

- مرتب کردن صفحه با استفاده از جدول (HTML)

اکنون که می توانید ASP ها را کنار هم بگذارید وکارهای نیمه پویا را آنها انجام دهید . محاورات درست از اطلاعات به دست می آیند واطلاعات درهر کجا که باشند باید مرتب شوند ولی اکثر اوقات درزندگی روزمره , به ترتیب نیستند , پس باید به صورت بانک اطلاعاتی درآیند.

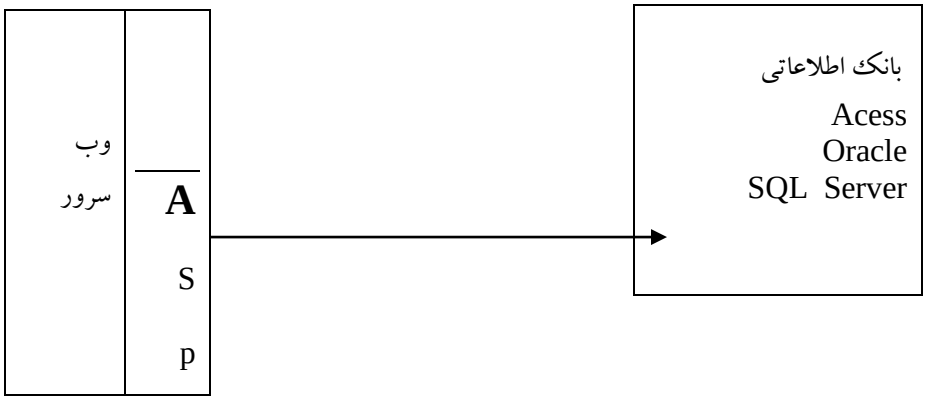
یک سوال مطرح شده این است که ازکدام بانک اطلاعاتی استفاده کنیم.

جواب یکی است: ((واقعا مهم نیست)).

3-8 انتخاب بانک اطلاعاتی و ODBC

تمام بانک های اطلاعاتی یک API (Application Programming Interface)هستند که برنامه نویسان برای گفتگو با آنها معمولا از C یا ++C استفاده می کنند . هرکدام با بقیه فرق دارد, دراین صورت, یک برنامه کاربردی که مستقیما با یک بانک اطلاعاتی مثل یک مترجم عمل ODBC گفتگو میکند نمیتواند با بانک اطلاعاتی دیگر گفتگو کند. ODBC مثل یک مترجم عمل میکند. برنامه نویس میتواند با ODBC API بنویسد و سپس ODBC آن فرامین را به بانک اطلاعاتی مخصوص ترجمه کند, با استفاده از یک درایور که مخصوص آن بانک اطلاعاتی است . به این طریق برنامه کاربردی از وقتی که درایور مناسب نصب شود, کارمیکند.

تا زمانی که این تئوری در مرحله یادگیری باقی بماند , هروقت تصمیم برتعیین معماری سیستم تولیدی خود می گیرید, اینکه کدام را انتخاب کنید اهمیت پیدا می کند. تمام بانکهای اطلاعات (یا درایورهای ODBC) ازهمه ویژگیها پشتیبانی نمیکنند وهیچکدام کاملا با استاندارد SQL مطابقت ندارند. ما به خاطر راحتی کار در اینجا, ازمایکروسافت اکسس استفاده خواهیم کرد ولی قبل از انتخاب یک سیستم برای برنامه کاربردی خود , ازدرستی انتخاب خود اطمینان حاصل کنیم, طوریکه ویژگیها وظرفیتی که شما لازم دارید, داشته باشد.



شکل ۱-۲ :ASP ها با استفاده از استانداردی مثل ODBC شما را درانتخاب یک بانک اطلاعاتی محافظت می کند .

دراین پروژه بانک اطلاعاتی به کاررفته , مایکروسافت اکسس می باشد , چون هم بکارگیری آن آسان است و هم به دست آوردن آن ساده می باشد. در واقع، اگر شما جداول واشیا, دیگر را با استفاده از SQL , درنظر ندارید حتما لازم نیست از طریق ASP با اکسس بانک اطلاعاتی بسازید و استفاده کنید.

بعضی ممکن است روی ماشین خود بانک اطلاعاتی دیگری رانصب کرده باشند مثل Oracle یا SQL server . تا وقتیکه ODBC دارید و به خوبی کار می کند برای اتصال به این بانک های اطلاعاتی مشکلی ندارید . برای انجام این کار , دستورالعملهای متعددی برحسب نوع سیستم ارائه خواهد شد پس باید مستندات ارائه شده را بررسی کنید. بعداز اتصال ASP به بانک اطلاعاتی , درروش دستیابی صفحات به آن , تفاوت کلی وجود ندارد

* توجه : تمامی بانک های اطلاعاتی از همه ویژگیهای موجود در ADO (Activex data Object) پشتیبانی نمی کنند پس به ازای دستیابی به آنها از OLEDB وODBC استفاده خواهیم کرد.

اگر مهمان یک ماشین خارجی شوید اطلاعات مبنی بر چگونگی دسترسی به بانک اطلاعاتی را به دست آورده اید, درغیراین صورت باید به مدیر (administrator) متصل شوید. بعضی کارها که درصدد انجام آن هستید بدون دسترسی به خود ماشین امکان پذیر نیست ولی شما برای کسب اطلاعات مناسب و مربوطه می توانید به آنها مراجعه کنید.

3-9ایجاد بانک اطلاعاتی و ODBC DSN :

پس از نصب مایکروسافت اکسس باید برای پروژه یک بانک اطلاعاتی ایجاد کنیم . ابتدا اکسس را باز کرده و یک بانک اطلاعاتی خالی به نام outpost. mdb بسازید توجه کنید که آنرا کجا ذخیره می کنید , چون مدتی به آن احتیاج دارید . وقتی این کار را انجام دادید از اکسس خارج شوید.

اکنون باید برای بانک اطلاعاتی که استفاده می کنید (DSN) Data Source Name ایجاد می کنید . یک DSN مثل مجموعه ای از اطلاعات پیکربندی ODBC برنامه کاربردی ماست که برای دسترسی به بانک اطلاعاتی استفاده می شود . برای انجام این عمل , به start/ settings/ control panel بروید و بر روی پوشه Administration Tools کلیک کنید . سپس روی نماد (ODBC) Data Source کلیک کنید . برگه System DSN را که بالای پنجره است , انتخاب کنید.

برای بانک اطلاعاتی خود , دراین حالت اکسس , درایور را انتخاب کنید , وروی Finish کلیک کنید در این مرحله , آماده انتخاب بانک اطلاعاتی واقعی خود هستید , پس نام outpost را به عنوان Data Source Name پر کنید . شرح (Description) می تواند هرچیزی که می خواهید باشد , در واقع Description اختیاری است روی select کلیک کرده و فایل outpost. mdb که دراکسس ساختید را پیدا کنید. اگر بانک اطلاعاتی نصب شده ندارید , روی Create کلیک کنید تا بانک اطلاعاتی اکسس را بسازد. چنانچه پیغام خطایی دریافت کردید, می گوید شاخه شناخته شده نیست (invalid) , مطمئن شوید که اکسس را بسته اید. حالا می توان بانک اطلاعاتی را برای استفاده در برنامه های کاربردی دیگر آزاد کند.

3-9-1درج رکوردها دربانک اطلاعاتی

ابتدا اکسس را باز کنید و بعد بانک اطلاعاتی outpost را باز کنید . اگرخطایی دریافت کردید مبنی بر اینکه برنامه کاربردی دیگری از بانک اطلاعاتی استفاده می کند و

نمی توانید دسترسی انحصاری داشته باشید , اشکالی ندارد , شما دیگر دسترسی انحصاری نمی خواهید , چون دوست دارید ASP هم قادر به استفاده ازآن باشد.

Product ID	Product Name	price	Description
3482	Troy VII	462000000	This planet..
6125	Rocket Jetpack	350	Soar high..
2165	Titus 3 Manual	22	Replaces..
9763	Taginos Greatest Hits	150	All of the..

شکل ۲-۲: یک جدول بانک اطلاعاتی دارای سطرها وستونهایی ازداده هاست.

اکنون فقط یکجدول به نام ad_log خواهید ساخت یک لحظه آن نام را درنظر بگیرید . اگر می خواستید می توانستید در نام جدول یک جای خالی قرار دهید و باز هم ازطریق ASP به آن دسترسی پیدا کنید. ولی این عمل استاندارد نیست و بعدا وقتی سایت موفق شد و به نتیجه رسید , شاید نخواهید اندازه بانک اطلاعاتی خود را بالا ببرید و به طور کلی ساختار و داده ها در بانک اطلاعاتی قویتری تکثیر کنید. احتمال دارد آن بانک اطلاعاتی از فاصله در اسامی جدول پشتیبانی نکند, پس در آن صورت باید تغییرات زیادی اعمال کنید . باید خود را برای یک مسئله قابل پیش بینی آماده کنید.

به یاد داشته باشید که اگر تا جایی که ممکن است استانداردها را تحمل کنید با مشکلات کمتری مواجه می شوید .

اگر تا به حال اکسس را بازنکرده اید, آن را باز نکنید , همچنین بانک اطلاعاتی outpost راروی New کلیک کنید تا یک جدول جدید را بسازید و Design View را انتخاب کنید.

درستون Field Name , کلمه Sponser را تایپ کنید. دکمه Tab را فشاردهید تا به ستون data Type حرکت کند . متن را انتخاب کنید وفیلد Size را به بیشترین مقدار تنظیم کنید (set) , که دراکسس, ۲۵۵ است . به این معنی که ستونی درجدول خود ایجاد می کنید و می توانید به اندازه یک متن ۲۵۵ کاراکتری را درآن قراردهید.

دوبارTab را کلیک کنید تا به ستون Field Name روی خط بعدی بروید. کلمه click Date را تایپ کنید و Date Type را به صورت date/time تنظیم کنید.

فکر کردن درباره نوع داده ها بسیارمهم است. VBScript یک زبان قوی ازنظر داده ها نیست , یعنی به یک متغیر هرنوع داده ای می توانید تخصیص دهید . ولی بانک اطلاعاتی به این صورت نیست. اگر به یک بانک اطلاعاتی بگویید که فیلدی به طول ۵۰ کاراکتر است ،۵۱ را قبول نمی کند.

پنجره را می بندید و جدول را با نام ad_log ذخیره کنید . ا کسس می پرسد که آیا می خواهید یک کلید اصلی (primary) برای این جدول تعیین کنید . روی No کلیک کنید.

3-9-2ایجاد اتصال به با نک اطلاعاتی

به دلایلی, مهمترین سازندهساخته شده در ASP , (Activex data object) که اشیائی از نوع COM می باشند و در ODBC استفاده می شوند و یا در OLEDB , برای دستیابی به بانکهای اطلاعاتی و ذخیره کردن اطلاعات دیگر. پس ADO ها زبان مستقلی هستند و نه تنها از طریق ASP بلکه از طریق ویژوال C++ و محیط های برنامه نویسی دیگر نیز می تواند مورد استفاده قرار گیرند.
ADO ها این امکان را به ما می دهند تا با بانک اطلاعاتی ارتباط برقرار نمائیم تا بتواند کارهای زیر را انجام دهد:

- Insert : درج داده های جدید به یک جدول.
- Update : عوض کردن داده هایی که ازقبل در جدول بوده اند.
- Delete : پاک کردن داده ها در جدول.
- Select : دیدن داده ها در جدول.

Execute Stored Procedure : بانک های اطلاعاتی پیشرفته تری هم وجود دارد که قابلیت کدنویسی و ذخیره کد به طور مستقیم در بانک اطلاعاتی را به ما می دهند که برای هر برنامه متصل شده ای (connected) قابل استفاده است.

به هرحال قبل از اینکه بخواهید هر اقدامی بکنید لازم است واقعا به بانک اطلاعاتی متصل شوید . برای انجام این عمل , یک شئ connection بسازید.

فایل go To Sponser. Asp را بازکرده و تغییرات زیر را به آن ا ضافه کنید.

```
<%Lnguage="VBSCRIPT "%>
<%

sponsorURL=Request.querystring("url)
```

```
creat the obgect
set outpost =server.cteateobject ("ADDOB.Connection")
open the connectin
outpost DBopen "outpost"

insert the record here


'close the conection
outpostDB.colose
'destory the conectin
set outpostDB=nothing

Reponse.redirect sponsorURL
%>
```

یک شئ connection را در خط ۵ ساخته اید , ولی تا وقتی که connection را بازنکرده اید هیچ کاری نمی توانید بکنید.

در اینجا همان DSN/outpost راباز کرده که قبلا ساخته اید. اگر به نام کاربری وکلمه عبور نیاز داشتید آنها را بعد از DSN با کاما جدا کرده و بنویسید

نکته : اگرسایت شما میزبان خارجی داشته باشد , ممکن است بدون تنظیم مدیر (Administrator) نتوانید از DSN استفاده کنید . دراین حالت , به اطلاعات خاصی که معمولا در DSN وجود دارند نیاز دارید, مثل نوع بانک اطلاعاتی و مکان فایل .

به عنوان مثال لیست زیر را ببینید:

برای اطلاعات بیشتر درباره DSNless سایت زیر را ببینید :

http: // support. Microsoft. Com / support/ kb/ articles/ q193 / 3/32.asp

3-9-3ساخت یک ارتباط DSNless

```
…
sponsorURL=Request.querystring("url)
```

```
creat the obgect
set outpost =server.cteateobject ("ADDOB.Connection")
open the connectin
outpost DB."open “driver={microsoft  Access Druver (*.mdb)};=c:\outpost.mdb
…
```

بازکردن یک connection تا وقتی که هدفی برای استفاده از آن نداریم, خیلی خوب نیست. بعدا اطلاعات را در بانک اطلاعاتی درج خواهیم کرد, مثل آگهی که روی آن کلیک شده است. ولی الان برای یادآوری خودتان یک توضیح (comment) را اضافه کنید تا بعدا بدانیم این , همانجایی است که باید مراجعه کنید.

وقتی که تمام تغییرات را اعمال کردید, connection را ببندید. این عمل به بانک اطلاعاتی می گوید که کار را تمام کرده اید . پس میتواند session را ببندید, درغیر این صورت بانک اطلاعاتی اتصالی دارد که هیچ کاری انجام نمی دهد ولی ازمنابع (resources) استفاده می کنند . گرچه این عمل, منابع روی وب سرور را آزاد نمی کند. برای آزادسازی حافظه با استفاده از شئ outpostDB لازم است متغیرها یا مراجع دیگر به این شئ را خراب کنید.

(destroy). این همان چیزی است که با برابر قرار دادن outpostDB مساوی Nothing انجام می دهیم. به این ترتیب وب سرور می داند که تمام متغیرهای مربوط به این شئ مساوی Nothing شده اند و می تواند ازمنا بع شئ برای چیزهای دیگر استفاده کند .

اکنون که می دانیم چگونه به بانک اطلاعاتی ارتباط برقرار کنیم , لازم است به زبان پرس وجوی ساختیافته SQL , زبان فرمانی که به ما اجازه می دهد تغییراتی مثل اضافه کردن, تعویض یا حذف داده ها را اعمال کنیم, نگاهی بیندازیم . SQL همچنین به ما اجازه بازیابی اطلاعات را می دهد. حال بیایید با درج داده ها شروع کنیم.

3-10مفا هیم مقدماتی SQL : درج کردن (Insert)

یکی از خوبیهای SQL این است که استاندارد می باشد . پس هر بانک اطلاعاتی با هرخصوصیتی , SQL را به فرم منحصر به فردی می سازد, تقریبا همه بانک های اطلاعاتی فرمانهای اولیه را متوجه می شوند . زیبایی دیگر SQL

این است که اگر به عنوان یک جمله هم به آن نگاه کنید خیلی ملموس است . درمسیر این پروژه شما چهارجمله SQL را می بینید: insert , Update, delete, select . ولی درحال حاضر فقط به insert توجه کنید .معمولترین فرم جمله insert در زیر آورده شده است.

insert into table name (columnnames) values (data values)

اکثر بانک های اطلاعاتی اسامی ستونها را تحت شرایط آنها قرار می دهند ، ولی اگر به این شکل عمل کنید بعدا دچار دردسر می شوید. پس URL و تاریخ جاری را درجدول ad log درج کرده و از فرمان زیر استفاده کنید.

Insert into ad log (sponsor , click Date) values (URL , Now)

این فرمان به بانک اطلاعاتی می گوید که به سراغ جدول ad log برود و یک رکورد جدید ایجاد کند و در ستون sponsor , مقدار آن برابر URL قراردهد و همچنین در ستون clickDate , زمان و تاریخ جاری را قرار دهد. برای اجرای یک فرمان ساده مثل این، از متد execute شئ connection استفاده کنید و با فرمان SQL یک رشته (string)، به آن بدهید. بیایید نگاهی به رشته های نشان داده شده در لیست پایین بیاندازیم:

```
<%Lnguage="VBSCRIPT "%>
<%

sponsorURL=Request.querystring("url)

creat the obgect
set outpost =server.cteateobject ("ADDOB.Connection")
open the connectin
outpost DBopen "outpost"

insert the record here
sqlText="insert into ad_log(sponser ,clickDate) values (‘"
sqlText= sqlText=&sponsorURL
sqlText= sqlText=& " ‘   ‘"
sqlText= sqlText=NOW
sqlText= sqlText=" ‘   ) "
```

Response.Write sqlText

```
'close the conection
outpostDB.colose
'destory the conectin
set outpostDB=nothing

Reponse.redirect sponsorURL
%>
```

قبل از اینکه به بانک اطلاعاتی نگاه کنیم نگاه سریعی به روش ساختن رشته متنی بیندازید : sqlText , تا بتوانید خروجی آن را ببینید.

درواقع sqlText از پنج قسمت تشکیل شده , بعضی از آنها متن ساده هستند و بعضی از آنها را از متغیرها دریافت میکنید. هنگام اجرای کد به عبارت زیر می رسیم:

insert into ad log (sponsor , clickDate) values
(‘http: // localhost/ ads/ intergal. Asp’, ‘ 7/26/99 10 : 52: 24 AM’)

متنی که پررنگ (bold) نوشته شده , ازمتغیرها به دست آمده است. اکنون که می دانید چه چیزی می خواهید , مانند لیست بالا آنرا به بانک اطلاعاتی بفرستید.

برای امتحان کردن این عمل , به http: // localhost/ index.asp بروید و روی آگهی banner کلیک کنید . سپس اکسس را باز کنید وجدول را نگاه کنید. رکورد مورد علاقه شما آنجاست.

این مثال خیلی ساده ای , ازآنچه می توانید با ADO انجام دهید, می باشد. همان طور که پیش می روید مثالهای پیشرفته تر زیادی را خواهید دید.

3-10-1درج داده ها

```
insert the record here
sqlText="insert into ad_log(sponser ,clickDate) values (‘"
sqlText= sqlText=&sponsorURL
sqlText= sqlText=& " ‘    ‘"
sqlText= sqlText=NOW
sqlText= sqlText=" ‘    ) "
outpostDB.Execute (sqlText)
```

```
'close the conection
outpostDB.colose
'destory the conectin
set outpostDB=nothing

Reponse.redirect sponsorURL
%>
```

3-10-2 فزودن خصوصیات محاوره ای (Interactive) به فرم ها

با وجود اینکه صفحه ها اساسا محاوره ای (Interactive) خوانده می شوند ولی در ابتدا , وب چندان intractive نبوده است. بیشترین مطلبی که راجع به آن می توان گفت این است که کاربران انتخاب می کردند که به کجا بروند, حتی اگردرباره اینکه چگونه به آنجا بروند , اطلاعی نداشتند. همچنان که همه چیز پیشرفت کرد, نویسندگان وب هم به دنبال روشی بودند که ازتجربه کاربران هم استفاده کنند . اکنون سایت ها, به طور خودکار , ردپای کاربران را دنبال می کنند, و سعی می کنند حدس بزنند چه چیزی برای آنها جالب است , پس می توانند مطالب را خیلی خوب به هم بدوزند! این فوق العاده است و قطعا جایگاه خاصی دارد . ولی می توانید راحت تر بفهمید که علایق کاربران چیست و می توانید ازآنها بپرسید.

یکی از ابداعات وب, فرم HTML می باشد که این مکان را به کاربر می دهد تا اطلاعات را به سایت وب ارائه دهد .

فرم, صفحه ای است که دارای اجزایی می باشد که به کاربر اجازه می دهد اطلاعات را ازطریق کادرهای متنی (text box) و دکمه های رادیویی (radio button) وارد کند و حداقل یک قسمت هم وجود دارد که به کاربر اجازه ارائه (submit) را می دهد.

این فرمها که با برچسبهای < form /></ FORM > مشخص شدند , می توانند به هرمنظوری استفاده شوند از درخواست کاربر , برای بازی گرفته تا دادن درخواستهایی به موتور جستجو . بعدا , شما فرمی خواهید ساخت که اطلاعات ثبت نام (registration) یک کاربر را می گیرد و اطلاعاتی راجع به انواع داستانهای علمی تخیلی را جمع آوری می کند.

3-10-3 گرفتن اطلاعات ازکاربران

کاربران نسبت به سایت هایی که اطلاعات شخصی را از آنها می خواهند , هوشیارتر می شوند. اگر بخواهید افراد را ثبت نام کنید (register), باید اعتماد آنها را به خود جلب کنید. یک روش جلب اعتماد این است که از یک جمله پنهانی که به آنها می گوید با اطلاعاتشان چه کاری می خواهید انجام دهید, استفاده کنید. مثل فروختن , اجاره دادن , یا قرض دادن اطلاعات به شرکتهای دیگر برای فرستادن لیست های آنها.

همچنین , تحقیقات نشان داده است که بهتر است هربار اطلاعات کمی را از کاربران جمع آوری کنیم. به جای اینکه از آنها بخواهیم که از اول , لیست های طولانی را پرکنند. در این صورت , اکثرا منصرف می شوند و از سایت کناره می گیرند.

فایل archives.asp را با کپی گرفتن از template.asp ایجاد کردیم . regisrter.asp را نیز به همان روش ایجاد خواهیم کرد. Template.asp را در ویرایشگر متن خود باز کنید و یک کپی به نام register.asp را در شاخه home ذخیره کنید. بعد یک فرم ساده با یک کادر متن مانند لیست زیر اضافه کنید.

3-10-4ساختن اطلاعات از فرم

```
<%Lnguage="VBSCRIPT "%>
<% pagetitle= Register  %>
<!. . #include virtual= “/ pagetop.txt”. .>

<H1>register <\H>
fill in the this form to become a member of primery outpost.
<p>
<form>
Desired Username:<input type=”text”>
</form>
<!. . #include virtual= “/ pageboton.txt. .>
</body>
</html>
```

اکثر مرورگرها وقتی کلید Enter را بزنید آن را ارائه می دهند (submit) .

امتحان کنید می بینید که کار خارق العاده ای انجام نمی دهد, فقط شما را به صفحه اصلی بر می گرداند . زیرا شما هیچ چیز دیگری به آن ندادید تا انجام دهد.

ابتدا لازم است برای فرم خود فایل مقصدی بسازید . template.asp را باز کنید و یک کپی به نام registration.asptake درشاخه home , روی وِب سِرور ذخیره کنید . چند خط متن به body اضافه کنید تا وقتی صفحه می آید, آنرا تشخیص دهید.

به register.asp برگردید برای اینکه بعد از ارائه فرم آن را به take registration.asp بفرستید یک Action مانند لیست زیر آن تخصیص دهید.

3-11افزودن یک Action

```
<%Lnguage="VBSCRIPT "%>
<% pagetitle= Register  %>
<!. . #include virtual= “/ pagetop.txt”. .>

<H1>register <\H>
fill in the this form to become a member of primery outpost.
<p>
<form  actionform ACTION= “/take registeration.aso”>
Desired Username:<input type=”text”>
</form>
<!. . #include virtual= “/ pageboton.txt. .>
</body>
</html>
```

در این کد به مرورگر می گویید که می خواهید وقتی فرم ارائه شد , به آن صفحه برود . این صفحه را ذخیره می کنیم آن رادر مرورگر خود بارگذاری کنید ودوباره امتحان کنید. اکنون به صفحه ای که به دنبالش می گشتید می روید.

3-12نامگذاری عنصر user name

```
<%Lnguage="VBSCRIPT "%>
<% pagetitle= Register  %>
<!. . #include virtual= “/ pagetop.txt”. .>

<H1>register <\H>
fill in the this form to become a member of primery outpost.
<p>
<form  actionform ACTION= “/take registeration.asp>
Desired Username:<input type=”text” NAME =”p_name” >
```

```
</form>
<!. . #include virtual= “/ pageboton.txt. .>
</body>
</html
```

فایل را ذخیره کنید و آن را بارگذاری مجدد نمایید و دوباره امتحان کنید . توجه کنید که URL به صورت زیر نشان داده می شود:

http: // localhost/ take registration.asp ?P name = My + text

این مزیت مهمی است , اقلامی که نام دارند, به همراه فرم ارائه می شوند (submitted) . اقلامی که نام ندارند , ارائه نمی شوند . اکنون بیایید دو کار انجام دهید . نگاه سریعی به بعضی ازصفات (attribute) کادرمتنی (text box) بیندازید اندازه را به بیشترین مقدارطول متن تنظیم کنید و یک مقدار اولیه برای اولین باری که فرا خوانی می شود , تخصیص دهید. هنگامی که این اجزا را اضافه کردید , دو دکمه هم اضافه کنید.

3-13ویژگی های فیلد متنی و افزودن دکمه ها

```
<%Lnguage="VBSCRIPT "%>
<% pagetitle= Register  %>
<!. . #include virtual= “/ pagetop.txt”. .>

<H1>register <\H>
fill in the this form to become a member of primery outpost.
<p>
<form  actionform ACTION= “/take registeration.asp>
Desired Username:<input type=”text” NAME =”p_name” size=20
MAXLENGTH=15
VALUE=”MyuserName”>
<P>
<Input type = “submit” value=“submit registration”><input type=”reset”
value=”start over”
</form>
<!. . #include virtual= “/ pageboton.txt. .>
</body>
</html>
```

به هیچ کدام از این دکمه ها نام نداده اید, چون فقط یک دکمه submit داریم، پس لازم نیست به آن نام اختصاص دهید. ولی مقادیر را به آنها اضافه کرده اید. شبیه کادر متنی, افزودن یک مقدار (value) به دکمه تعیین می کند که وقتی شما صفحه را می سازید , روی دکمه چه چیزی نوشته شود. برخلاف یک کادر متنی, نمی توانید مقدار روی دکمه را عوض کنید.

به کادر متنی ساخته شده نگاه کنید می بینید که مقدار بعضی چیزها را به صورت آنچه بیشتردوست دارید تغییر دهید. قابل توجه که بیشتر از ۱۵ کاراکتر را در کادر نمی توانید جای دهید . چون خودمان maxlength رااین طور تنظیم کرده اید. این مقدار , کاملا دستی است. مخصوصا وقتی که داده ها را نهایتا داخل بانک اطلاعاتی می گذارید . یادتان باشد که اگر مشخص کردید طول کاراکترها ۵۰ باشد , ۵۰ آخرین مقداری است که فراهم می شود . بهتر است وقتی کاربران کاراکترهای بیشتر از حد مجاز خود را از طریق فرم وارد می کنند , کارمتوقف شود وکاربران به مرحله قبل فرستاده شوند. چون اطلاعات آنها مناسب نبوده است.

احتمالا دکمه های Reset را که برای یک فرم استفاده می شوند برروی وب دیده اید . اکثر افراد متوجه نمی شوند که پاک کردن فرم کا ملا آن طور که دکمه Reset انجام می دهد , نیست. دکمه Start over (Reset) را کلیک کنید . به جای پاک کردن فرم, کار موثرتری را انجام می دهد ، یعنی به حالت اولیه فرم برمی گردد.

درپایان , دکمه submit است که کاملا برحسب آنکه چه توقعی ازآن داشته باشید عمل می کند . ممکن است درحال حاضر به آن نیاز نداشته باشید چون فقط یک فیلد متنی دارید. ولی به محض اینکه یکی دیگر اضافه کنید, دیگر مرورگر درصورت فشردن Enter , آنرا ارائه نمی کند (sunmit).

بخش اول از صفحه ثبت نام (registration) خودرا اضافه کنید.

از کادرهای متنی همه جا استفاده کنید بجز برای دو ورودی کلمه عبور. کادر کلمه عبور (password) کاملا شبیه کادر متنی است . بجز اینکه هنگام تایپ کردن درآن, حروف را نمی توانید ببینید. این کار برای جلوگیری کردن از این است که شخص دیگری نتواند به آن نگاه کند وکلمه عبور را هنگام برقراری ارتباط از روی دست ما بدزدد.

3-14افزودن اطلاعات شخصی

```
<%Lnguage="VBSCRIPT "%>
<% pagetitle= Register  %>
<!. . #include virtual= “/ pagetop.txt”. .>

<H1>register <\H>
fill in the this form to become a member of primery outpost.
<p>
<form  actionform ACTION= “/take registeration.asp>
```

```
Desired Username:<input type=”text” NAME =”p_name” size=20 MAXLENGTH=15>
<BR>
PASSWORD:<INPUT TYPE=”password name=”   NAME= “p_pass”1>
<BR>

PASSWORD  (again):<input type=” password” NAME =”P_pass2”>
<BR>
FIRST NAME: <input type =”text” name=”p_first”>
<BR>
LAST NAME :<INPUT TYPE=;TEXT” NAME=”p_last”>
<BR>
EMAIL ADDRESS:<INPUT TYPE=”text” NAME=”p_email”>
<P>
<Input type = “submit” value=“submit registration”><input type=”reset” value=”start over”
</form>
<!. . #include virtual= “/ pageboton.txt. .>
</body>
</html>
```

اکنون یک فرم ابتدایی دارید حال کارهایی روی آن انجام دهید .

فایل take registration.asp خود را باز کنید. دراین حالت دو کار را انجام می دهید . اول اینکه مقادیر (values) را بگیرید و در متغیرها بگذارید و آنها را روی صفحه چاپ کنید تا ببینید .

ممکن است تعجب آور باشد که وقتی داده ها را نهایتا در بانک اطلاعاتی می گذارید , این کار را انجام داده اید جواب آن اینجاست : اگر شما یک خانه بسازید , آیا اول همه چیز را می چینید , بعد از انجام همه این کارها , ازاینکه فونداسیون آن خوب است مطمئن می شوید؟ البته که اینطور نیست ! در عوض , هرمرحله ای را که طی می کنید , بررسی کنید تا مطمئن شوید که آن مرحله قبل از هراقدام دیگری , درست کار می کند . دراینجا هم همین طور است. برای رفع هر گونه مسئله ای , فقط باید دریک یا دو مرحله نشانه ای بگذارید تا بعدا بدانید که کجا بودید. فایل take registration.asp را بازکنید , شما قبلا با ویرایشگر متن آن را ساخته اید .

3-15 گرفتن اطلاعات از فرم

```
<%Lnguage="VBSCRIPT "%>
<!. . #include virtual= “/ pagetop.txt”. .>
<%
```

```
p_user id=request.querystring("p_p-name")
      p_pass1=request querystring("p_pass1")
      p_pass2=request querystring("p_pass2")
      p_first=request querystring("p_first")
      p_last=request querystring("p_last")
      p_email=request querystring("p_email")

%>

<H2>user registration</H2>
user id : <%=p_userid%^><BR>
password:<%=p_pass%><br>
password (again):<%  ="p_pass2  %><BR>
first  Name : <%  ="p_first  %><BR>
last Name :  <%  ="p_last  %><BR>
Email     :   <%  ="p_mail  %><BR>

<!. . #include virtual= "/ pagetop.txt". .>
</BODY>
</HTML>
```

هنگامی که فرم را ارائه کردید (submit) , لیستی از اجزای فرم را می بینید.

حالا نوع دیگری از اجزای فرم مانند یک دکمه رادیویی (radio button) را به فایل register.asp اضافه کنید. دکمه های رادیویی , بسیار مفید هستند چون اطمینان می دهند که کاربران فقط یک مقدار وارد می کنند . پس انتخابهای هرکدام منحصر به فرد است. یک سوال بله یا خیر اضافه کنید.

3-16اضافه کردن دکمه رادیوئی

```
EMAIL ADDRESS:<INPUT TYPE="text" NAME="p_email">
<BR>
Do you belive in alines?
<input type=" RADIO NAME =P-alines" VALUE=" yes"checked>Absuoltly
< input type=" RADIO NAME =P-alines" VALUE=" no">don't be ridiculous
<P>
<input type  ="radio" NAME="p_medium"  VALUE="submit registration".<in
put type="reset" value="start over"
</form>
```

```
<!. . #include virtual= “/ pageboton.txt. .>
</body>
</html>
```

اگربه خط های

```
<input type=” RADIO NAME =P-alines” VALUE=” yes”checked>Absuoltly
 input type=” RADIO NAME =P-alines” VALUE=” no”>don’t be ridiculous
```

توجهکنید که صریحا مقداری را به دکمه رادیویی تخصیص داد که با آنچه که شما واقعا درصفحه می بینید , مطابقت نمی کند . این مقدار, با فرم ارائه می شود .متن روی صفحه یا برچسب برای کمک کردن به کاربر است وفرم به آن توجهی نمی کند. می توانستید جوابها را تغییر دهید , فرم هیچ وقت درباره آن چیزی نمی داند مثلا ارائه کردن مقدار N0 وقتی که کاربر Absolutely را انتخاب کرده یا برعکس.

نام واقعی این , دکمه رادیویی نیست , بلکه نام آن گروه رادیویی می باشد . دلیل آن این است که فرم آنها را براساس نامشان (Name) گروهبندی می کند تا مطمئن شودکه فقط یک گروه در هر زمان دریافت می شود.

اکنون دسته دومی از دکمه ها درباره علایق کاربران سوال می کنند.

3-17اضافه کردن یک گروه رادیوئی دیگر

```
< input type=” RADIO NAME =P-alines” VALUE=” no”>don’t be ridiculous
<P>
How do you enjoy your siens fiction?<BR>
<input type  =”radio” NAME=”p_medium”  VALUE=”TV”>TV<BR>
<input type  =”radio” NAME=”p_medium”  VALUE=’movies”>movis<BR>
<input type  =”radio” NAME=”p_medium”  VALUE=”books” >books<BR>
<input type  =”radio” NAME=”p_medium”  VALUE=”comies”>comies<BR>
<input type  =”radio” NAME=”p_medium”  VALUE=”online”>online<BR>
<input type  =”radio” NAME=”p_medium”  VALUE=”fanzins”>fanzins<BR>
<P>
<Input type = “submit” value=“submit registration”><input type=”reset”
value=”start over”
</form>
<!. . #include virtual= “/ pageboton.txt. .>
</body>
</html>
```

توجه کنید که چون از اول دکمه ای را به صورت CHECKED, تنظیم نکرده اید , هیچکدام ازآنها علامت نخورده اند . به هرحال به محض اینکه دکمه ای کلیک شود , حداقل یک گزینه برای ارائه پذیرفته می شود و دیگر راهی برای انتخاب نشدن آن عنصر وجود ندارد, مگر با انتخاب کردن یکی دیگر از اجزای گروه.
مشکل یک گروه رادیویی این است که شما فقط یک عنصر علامت خورده دارید و چون می خواهید به کاربران اجازه دهید چندین گزینه را علامت بزنند, می توانید این کار را با تغییر دسته دوم دکمه ها ی رادیویی به کادرهای انتخاب (check box) انجام دهید.

3-18استفاده از کادرهای انتخاب بجای گروه رادیوئی

```
<P>
How do you enjoy your siens fiction?<BR>
<input type  =”checkbox”NAME=”p_medium”  VALUE=”TV”>TV<BR>
<inputtype=”checkbox”radio”NAME=”p_medium”VALUE=’movies”>movis
<BR>
<input type=”checkbox” NAME=”p_medium”  VALUE=”books” >books<BR>
<inputtype=”checkboxradio”NAME=”p_medium”VALUE=”comies”>comies
<BR>
<inputtype=” checkbox” NAME=”p_medium”  VALUE=”online”>online<BR>
<inputtype=”checkbox”NAME=”p_medium” VALUE=”fanzins”>fanzins<BR>
<P>
<Input type = “submit” value=“submit registration”><input type=”reset”
value=”start over”
</form>
<!. . #include virtual= “/ pageboton.txt. .>
</body>
</html>
```

کاربران با کادرهای انتخاب می توانند گزینه های مختلفی را که دوست دارند, انتخاب کنند و این باعث به وجودآمدن نتایج کاملا جدیدی می شود .
اقلام medium متعددی را انتخاب کنید و فرم را ارائه کنید (submit) . به URL ، روی مرورگر نگاه کنید . چند بار p medium را با مقادیر مختلف خواهید دید. وقتیکه ASP می کوشد querystring را مقداردهی کند و مقداری را برای p medium بیرون بکشد. درواقع مجموعه ای از مقادیر , یک نوع آرایه به دست می آورد.

p- medium
Movies
Television
Books
Online

شکل۳-۲: یک آرایه , یک سری شماره از اقلام مربوط به هم است.

به جای نمایش دادن اینکه چه تعداد از اقلام در آرایه وجود دارد وحلقه گذاشتن درآن , به عهده ASP بگذارید . همانطور که در لیستزیر آورده شده است.

استفاده از کادرهای انتخاب بجای گروه رادیوئی

```
<H2>user registration</H2>
user id : <%=p_userid%^><BR>
password:<%=p_pass%><br>
password (again):<%  =”p_pass2  %><BR>
first  Name : <%  =”p_first  %><BR>
last Name :  <%  =”p_last  %><BR>
Email     :   <%  =”p_mail  %><BR>
<%
for each p_mdium in request .querystring(“p_medium)
%>
perferred medium : <%= p_medium %><BR>
<%
next
%>
<!. . #include virtual= “/ pagetop.txt”. .>
</BODY>
</HTML>
```

کد VBScript با یک حلقه for each بین اقلام حلقه می زند و هر قلم را به p medium تخصیص می دهد. در اینجا , قلم باید به صورت متن باشد (ویک نوع شئ) هروقت قلم دیگری به صورت علامت نخورده نبود حلقه به آخر می رسد. دراین روش مهم نیست که چند جز را یک کاربر انتخاب می کند , باز هم کار خواهد کرد.
اکنون شما همه اطلاعات را دارید , حالا باید درباره اینکه چگونه این اطلاعات را دربانک اطلاعاتی ذخیره کنید, تصمیم بگیرید. این پردازه مدل سازی داده ها نام دارد.

3-18مدل سازی داده ها (Data Modeling)

یک نکته درباره ساختن سایتهای وِب , این است که شما با مسائل کاملا متفاوتی برخورد می کنید. خوشبختانه نباید درباره تمام زمینه ها تجربه داشته باشید. چون مدل سازی داده ها مفهوم دیگری است که خودش می توانست کتابی باشد . در طراحی یک بانک اطلاعاتی عوامل متعددی را باید در نظر گرفت . ازتصمیم گیری نوع بانک اطلاعاتی که در ابتدا استفاده می شود گرفته تا روش پخش کردن فایل ها برروی دیسک و اینکه چه جداولی باید به کاررود . به هرحال , دراین جا, شما بانک اطلاعاتی بسیار کوچکی دارید , پس می توانید مدل سازی را نادیده بگیرید ، ولی جداول موجود هستند.

توجه: دوکتاب درزمینه مدل سازی داده ها معرفی می شود : *Entity Relationship Modeling* (*Computer Aided Systems Engineering*) نوشته *Richard Barker* وهمچنین کتابی دیگر با عنوان *Function and Process Modeling* نوشته *Richard Barker* و *Cliff Longman*

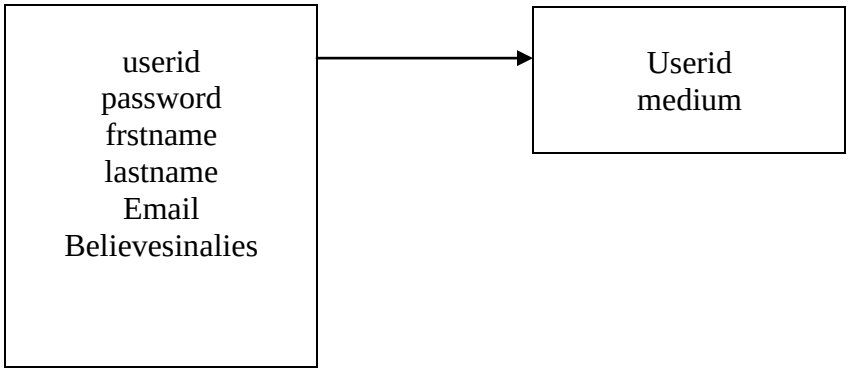
معمولا , ابتدا بانک اطلاعاتی را کامل طراحی می کنید . اگر یک بخش آن بر بقیه تاثیر بگذارد, لازم نیست بعدا بانک اطلاعاتی را تغییر دهید ,برای اینکه دراین پروژه همه چیز را ساده در نظر بگیرید . هربخش را تا حدی که لازم است طراحی کنید.

3-18-1تعیین موجودیتها (Entities)

موجودیت (entity) مسئله مهمی است باید پیگیری شود . اکنون به سراغ کاربران برویم، اطلاعاتی از آنها بگیرید . دقیقا به دنبال چه اطلاعاتی از کاربران هستیم.
اگر می خواهید ستون دیگری به آن جدول بیفزایید، برای نگهداری تمام موارد, فضای هدررفته زیادی دارید که کاربر برای پاسخ دادن از آن استفاده نمی کند . بجای این ,کاربران وعلایق (preferences) آنها را در موجودیتهای متفاوتی که به هم وابسته هستند, درنظر بگیرید .

می توانید فضای خالی هدررفته و مشکل اضافه شدن ستونهای زیاد را با لیست کردن آنها حذف و برطرف کنید. ولی وقتی لازم باشد بدانید چه تعداد ازکاربران به یک چیزعلاقه دارند , با چند بار لیست کردن اطلاعات کاربران, فضای به مراتب بیشتری هدر می رود . می توانید فقط با دو جدول این عمل را نیز انجام دهید.

Membersuserid- medium



شکل ٤-٢ : با استفاده از دو جدول و اتصال آنها به هم با یک کلید , مانند Userid , می توانید انعطاف پذیری زیادی ایجاد کنیم .

دراین وضعیت ، از نام کاربری (Username) به عنوان کلیدی که اطلاعات را به هم متصل می کند استفاده کنید.

3-18-2 **کلید اصلی (primary key)** :

درجدول member, username یک کلید اصلی می باشد که نشان دهنده این است که هرعضو دارای یک username بوده و هر username منحصر به فرد می باشد. اگراطلاعات یک عضو (member) رابخواهید می توانید به بانک اطلاعاتی بگویید دنبال username بگردد و به اطلاعات دیگری احتیاج نیست .

درمورد جدول User members , username یک کلید خارجی خواهد بود . یعنی اگر مقداری را برای username بگذارید , یک اتصال به جدول members ازقبل وجود داشته است.

یک نکته مهم درباره نامگذاری این است که , باید همخوانی داشته باشد اگر درباره نامگذاری کمی فکرکنید واقعا مهم نیست که قرارداد نامگذاری چه باشد. ولی خوب است که دیگران هم بدانند مبنای روش نامگذاری شما چیست؟ به همین دلیل جدول دوم را User medium نامیدید .یک نگاه سریع بیندازید . اکنون نه تنها می دانید که این جدول , چگونه جدولی می باشد , بلکه همچنین می دانید به چه جدول دیگری مربوط می شود.

3-18-3جدول واقعی:

اکنون که می دانید چه چیزی را ضبط می کنید و چگونه این کار را انجام می دهید لازم است طراحی جدول را به پایان برسانید می دانید که چه ستونهایی درهر جدول به کارمی رود حال باید تصمیم بگیرید که نوع داده هایی که درآنها می روند چیست و چقدر هستند. اسکریپت های سازنده برای این دوجدول به صورت زیر است:

```
Create table member(
User name  char(30)
Password  char (30)
First_name   char(30)
Last_name    char(30)
Email   char(30)
Comments   char(30)
Belives_in_alines  char(30)
create table user _medium(30)
username  char(30)
medium char(30)
```

یک اسکریپت سازنده , به ما اجازه می دهد و بدون داشتن هیچ ابزارخاصی بتوانید یک شی بانک اطلاعاتی , مثل یک جدول , را بسازیم آنها هرجایی که بتوان SQL را اجرا کرد می توانند اجرا شوند ودرصورت نیاز امکان دوباره سازی جداول را , به آسانی فراهم می کند.

فرض کنید که تمام ستونهای جدول , متن دارند و اغلب آنها به طول ۳۰ کاراکتر هستند , این بیشتر ازنیاز واقعی شماست ولی درحال حاضر آنرا بپذیرید.

در بانک اطلاعاتی , دوجدول به همان روشی که جدول ad log را ساختید ,ساخته می شوند . اگر بانک اطلاعاتی نصب شده روی سیستم شما ندارید و درهنگام نصب DSN , یک بانک اطلاعاتی اکسس ساخته اید می توانید فرمان execute را به کار برید تا ساختن اسکرییپت ها را اجرا کند وجدول شما ساخته شوند.

3-18-4درج داده ها:

اکنون که جداول را داریم , می توانید داده های member را وارد کنید .کاملا شبیه همان روشی که برای Ad log ها انجام دادید در لیست زیر:

```
<%Lnguage="VBSCRIPT "%>
<!. . #include virtual= “/ pagetop.txt”. .>
<%

p_user id=request.querystring(“p_p-name”)
```

```
    p_pass1=request querystring("p_pass1")
    p_pass2=request querystring("p_pass2")
    p_first=request querystring("p_first")
    p_last=request querystring("p_last")
    p_email=request querystring("p_email")

set outpostDB= server .createObject ( "ADDOB.Connection")

the SQL =  "insert into member "
the SQL= the SQL &"(username , password, first_name , last_name,"
the SQL= the SQL  & " email , believe_in_alines)
the SQL= the SQL  & "values('"&p_userid"', "&p_pass1&"', '"
the SQL= the SQL  &p_first& "',' "p_last" ' ' "p_email" ', ' "
the SQL= the SQL  &p_belives_in_in_alines&" ')'

outpostDB.Execute(the SQL)

for each p_medium in request .querystring( "p_medium"0
the SQL=" insert into userid_medium (userid , mesum) values
( " "& p_userid&" ; ,  ' "&p_medium&' ")

Outpost DB.Execute(theSQL)
Next

OutpostDB.close
Set outpost =nothing

 %>
<H2> user Registeration<H2>
```

اکنون فرم عضویت را پرکنید آن را ارائه دهید (submit) وجداول خود را بررسی کنید. ظاهرا کارفرم ثبت نام (registration) راانجام دادید , درصورتیکه اشتباه کنید، اگر صفحه را بارگذاری مجدد بکنید , یک کپی دیگرازاطلاعات عضویت خود را درجداول به وجود آورید ولی شما نمی خواهید همان نام کاربری بیش از یک داشته باشد به دو روش می توانید به این مسئله رسیدگی کنید . یک روش آسان , بررسی کردن تکراریها قبل از درج (insert) است ولی این روش فوق العاده کند خواهد بود.

روش دیگر این است که ستون Username را به عنوان کلید اصلی (primary) در بانک اطلاعاتی برگزینیم.

ستون Username را درجدول members به عنوان یک کلید اصلی قرار دهید . اگر در اکسس هستید روی ستون و در بالای آن کلیک کنید , درجایی که Indexes نوشته شده
(No Duplicated) Yesراانتخاب کنید . کنار Required , دکمه Yes راانتخاب کنید
(تمام اینها یعنی ایجاد یک کلید اصلی واقعی که منحصر به فرد است و خالی نیست.)
بعد از انجام این کار , بانک اطلاعاتی اجازه تکراری شدن رکوردها را نمی دهد . می توانید با درج رکورد تکراری آن را امتحان کنید وخطایش را بررسی کنید.

3-19رسیدگی به خطا ها :

ASP اشیاء اسکریپت متعددی را با وظایف مختلفی , برای کمک کردن به شما فراهم می کند. یکی ازآنها شئ Err است که دررسیدگی به خطاها کمک می کند.
اگر یکبار بعد ازاینکه صفحه خود را ارائه کردید (submit) , آن را بارگذاری کنید , پیغام خطایی دریافت خواهید کرد . این پیغام اطلاعات زیادی به ما می دهد و مطمئنا چیزی نیست که دوست داشته باشید کاربر آن را ببیند . به هرحال , می توانید خود به این پیغام رسیدگی کنید و از نمایش خطای صفحه اجتناب کنید.
با یک عبارت on Error می توانید به خطا رسیدگی کنید. به این معنی که اگر خطایی رخ دهد به ASP می گویید چه کاری انجام دهد . در VBScript عبارات رسیدگی به خطا , زیاد نیستند ولی با استفاده از OnError Resume Next می توان بر این قابلیت افزود . این عبارت به ASP می گوید که هر وقت خطایی دریافت کرد , باز هم به راه خود ادامه دهد. این تابع را به صفحه خود اضافه کنید.

3-19-1درج داده ها

```
<%Lnguage="VBSCRIPT "%>
<!. . #include virtual= “/ pagetop.txt”. .>
<%
on  Error  reseume next
p_user id=request.querystring(“p_p-name”)
     p_pass1=request querystring(“p_pass1”)
     p_pass2=request querystring(“p_pass2”)
```

اگر بارگذاری مجدد کنید , خطایی دریافت نخواهید کرد ولی هیچ نشانه و علامتی هم مبنی بر وجود خطا نخواهید گرفت . بنابر این , اکنون که خطایی را ازدست ASP گرفته اید , باید خودتان آن را بررسی کنید.

وقتی خطایی رخ می دهد , سه بخش اطلاعاتی درشی Err ثبت می شود, شماره خطا وتوضیحی ازخطا و منبع خطا , اگرخطایی وجود نداشته باشد , شماره خطا صفر است , پس عمل درج کردن می تواند انجام شود. پس این مقدار را بررسی کنید و ببینید که آیا عملیات موفقیت آمیز است یا خیر؟

3-19-2بررسی کردن خطاها

```
For each p_medium in request.querystring(*p_medium*)
The SQL=”insert into userid_medium(userid, medum0 values
( “ “& p_userid&” ; ,  ‘ “&p_medium&’ “)
Outpost DB.Execute(theSQL)
Next

OutpostDB.close
Set outpost =nothing

 %>
<% if Err.number  =0 then

        ‘All is well with the world%>

<H2> user Registeration<H2>
thankyou for registering with primery Outpost!
<% else %>

there was a problem with their registeration
<h2>problem</h2>
there was a problem with your registration
please go back and choose a differend user name.
<% end if %>

<!. . #include virtual= “/ pagebotton.txt”. .>
</body.
</html>
```

خط <% if Err.number =0 then , وجود خطا را بررسی می کند . سپس پیغام مناسب را نشان می دهد. قبل از اینکه خطاها را بررسی کنید , مطمئن شوید که کار بر پیش ازاینکه اجازه ثبت نام (register) را بگیرد , کلمه عبورخود را اشتباه تایپ نکرده باشد. می توان این عمل را با توجه به خط دوم عبارت if , then , else درخطوط در لیست زیر انجام داد.

3-19-3اطمینان از کلمه عبور

```
<%Lnguage="VBSCRIPT "%>
<!. . #include virtual= “/ pagetop.txt”. .>
<%

p_user id=request.querystring(“p_p-name”)
      p_pass1=request querystring(“p_pass1”)
      p_pass2=request querystring(“p_pass2”)
      p_first=request querystring(“p_first”)
      p_last=request querystring(“p_last”)
      p_email=request querystring(“p_email”)

set outpostDB= server .createObject ( “ADDOB.Connection”)

the SQL =  “insert into member “
the SQL= the SQL &”(username , password, first_name , last_name,”
the SQL= the SQL  & “ email , believe_in_alines)
the SQL= the SQL  & “values(‘”&p_userid”’, “&p_pass1&”’, ‘”
the SQL= the SQL  &p_first& “’,’ “p_last” ‘ ‘ “p_email” ‘, ‘ “
the SQL= the SQL  &p_belives_in_in_alines&” ‘)’

outpostDB.Execute(the SQL)

for each p_medium in request .querystring( “p_medium”0
the SQL=” insert into userid_medium (userid , mesum) values
( “ “& p_userid&” ; ,  ‘ “&p_medium&’ “)

Outpost DB.Execute(theSQL)
Next

OutpostDB.close
Set outpost =nothing


<% if Err.number  =0 then

          ‘All is well with the world%>

<H2> user Registeration<H2>
thankyou for registering with primery Outpost!

www.Prozhe.com
```

```
<% else %>

there was a problem with their registeration
<h2>problem</h2>
there was a problem with your registration
please go back and choose a differend user name.
<% end if %>

else
        'p_pass1 dosent match p_pass
<h2>password error </h2>
both entied for your password must match
end   if
<!. . #include virtual= "/ pagebotton.txt". .>
</body.
</html>
```

توجه کنید که عبارت if – then تودرتو است . یعنی اینکه شما عباراتی دارید که یکی داخل دیگری است . این ترکیب کاملا قابل قبول است و روش خوبی برای توضیح کدها و مرتب نگه داشتن آنهاست . موارد کمی هستند که از یک پیغام خطا مخربتر هستند و نمی توانید که این خطاها از کجا ناشی شده اند , چون کد ما هوشمند نیست.

3-20تکمیل فرم :

فرم تقریبا کامل است , ولی باز هم توجه کنید که تمام اطلاعاتی که ارائه شده است , ازجمله کلمه عبور , درخود URL می باشد. برای این مثال , بسیار خوب است , چون باید بدانید فرم چه کار می کند , ولی در یک برنامه کاربردی واقعی , این نظر خوبی نیست. آنچه باید انجام داد, تغییر متدی است که فرم با آن متد اطلاعاتش را به سِرور فرستاده است.

دو متد در دسترس ما است : post , get

وقتی یک فرم با استفاده از get , ارائه شود (submit) , اطلاعات را در querystring قرار می دهد . هر وقت یک فرم با استفاده از post ارائه شود , اطلاعات خودشان به request اضافه می شوند و به روشهای مختلفی پردازش می گردد.

اگرمشخص نشده باشد که باید از متداولی استفاده شود یا دومی , مرورگر از get استفاده می کند و get همان متدی است که قبلا با آن سروکار داشته ایم . شما از مرورگر در post استفاده می کنید. این قسمت را به برچسب <form> در register .asp اضافه کنیم . دراین بخش register.asp لیست زیرنشان داده شده است .

3-20-1تغییر دادن فرم از get به post

```
<%Lnguage="VBSCRIPT "%>
<% pagetitle= Register  %>
<!. . #include virtual= “/ pagetop.txt”. .>

<H1>register <\H>
fill in the this form to become a member of primery outpost.
<p>
<form action =”/take_registration.asp” metod=”post”>
desiered user name :< INPUT TYPE =”text” name =”p_name”
size= 20     maxlenght=15>
```

تمام بخشی که کاربر می بیند, همین بود. ولی هنوز هم باید در پشت کار , اصلاحاتی را انجام دهید. یعنی در همان بخش که ازدید کاربران مخفی می باشد . هنگامی که یک فرم با استفاده از post ارائه می شود , به URL اضافه نمی گردد, پس querystring وجود ندارد درعوض اطلاعات داخل مجموعه form قرار می گیرند. درنتیجه به جای استفاده از:

Request .querystring (“username”)

باید از عبارت زیراستفاده کرد :

Request .form (“ username”)

می توانید برای تغییر هر نمونه ای از querystring به form از ویرایشگر متن استفاده کنید,

بنابر این takeregistration.asp شبیه لیست زیر می شود.

3-20-2فرم نهائی

```
<%Lnguage="VBSCRIPT "%>
<!. . #include virtual= “/ pagetop.txt”. .>
<%

p_user id=request.querystring(“p_p-name”)
     p_pass1=request querystring(“p_pass1”)
     p_pass2=request querystring(“p_pass2”)
     p_first=request querystring(“p_first”)
     p_last=request querystring(“p_last”)
     p_email=request querystring(“p_email”)
```

```
if p_pass  = p_pass2 then

set outpostDB= server .createObject ( “ADDOB.Connection”)

the SQL =  “insert into member “
the SQL= the SQL &”(username , password, first_name , last_name,”
the SQL= the SQL  & “ email , believe_in_alines)
the SQL= the SQL  & “values(‘”&p_userid”’, “&p_pass1&”’, ‘”
the SQL= the SQL  &p_first& “’,’ “p_last” ‘ ‘ “p_email” ‘, ‘ “
the SQL= the SQL  &p_belives_in_in_alines&” ‘)’

outpostDB.Execute(the SQL)

for each p_medium in request .querystring( “p_medium”0
the SQL=” insert into userid_medium (userid , mesum) values
( “ “& p_userid&” ; ,  ‘ “&p_medium&’ “)

Outpost DB.Execute(theSQL)
Next

OutpostDB.close
Set outpost =nothing


<% if Err.number  =0 then

        ‘All is well with the world%>

<H2> user Registeration<H2>
thankyou for registering with primery Outpost!
<% else %>

there was a problem with their registeration
<h2>problem</h2>
there was a problem with your registration
please go back and choose a differend user name.
<% end if %>

else
        ‘p_pass1 dosent match p_pass
<h2>password error </h2>
both entied for your password must match
end   if
%>
```

```
<!. . #include virtual= “/ pagebotton.txt”. .>
</body.
</html>
```

فایل را ذخیره کنیم وآنرا امتحان کنیم. توجه داشته باشید که بعداز این, همه اطلاعات را در URL نداریم. درعوض خودش به عنوان بخشی از request فرستاده می شود.

در عمل بهتر است تا جایی که ممکن است از post استفاده کنیم . post به تنهای مطمئن ترین روش نیست (چون اطلاعات , به این سادگی نیستند.) ولی باز هم مطمئن تر است . اگر برای فرستادن حجم زیادی ازاطلاعات , از get استفاده کنیم , مشکلات زیادی بروز خواهد کرد.

به هرحال باید در نظرداشت که چون اطلاعات در URL نیستند , کاربر نمی تواند صفحه را محل یابی (bookmark) کند. اگربخواهید این کارانجام شود بایستی از get استفاده کرد.

منابع :

C# How To Program : Introducing .Net & Web Services

Deitel & Deitel

Prentice Hall Publication Inc.

Copyrights © 2002 Prentice Hall, Inc.

Mastering C# Database Programming

Jason Price

Sybex Publications

Copyrights © 2003 Sybex

http://csharp-persian.netfirms.com/

کتاب آموزش جامع سی شارپ اسلام احمد زاده

Learning ASP WROX